

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky

# OOP4FUN: OBJECT ORIENTED PROGRAMMING FOR FUN

Průvodce pro současné učitele středních škol k výuce programování

Editoři

Michal Varga, Josef Rak, Dušan Savić, Zlatko Stapić



Autoři:

Peter Sedláček, Nika Kvaššayová, Jozef Kostolný, Michal Mrena, Patrik Rusnák, Peter Sobe, Ilija Antović, Miloš Milić, Tatjana Stojanović, Davor Fodrek, Lidija Kozina, Marko Mijač, Dijana Plantak Vukovac, Antonela Čižmešija, Dijana Peras, Goran Hajdin, Lea Masnec

Pardubice, září 2024.



Co-funded by the  
Erasmus+ Programme  
of the European Union

Projekt Object Oriented Programming for Fun

Zkratka projektu OOP4FUN

Číslo projektu 2021-1-SK01-KA220-SCH-00027903

Koordinátor projektu Žilinská univerzita v Žiline (Slovakia)

Řešitelé projektu

- Sveučilište u Zagrebu (Croatia)
- Srednjaškola Ivanec (Croatia)
- Univerzita Pardubice (Czech Republic)
- Gymnázium, Pardubice, Dašická 1083 (Czech Republic)
- ObchodnaakademiaPovazskaBystrica (Slovakia)
- Hochschule fuer Technik und Wirtschaft Dresden (Germany)
- Gymnasium Dresden-Plauen (Germany)
- Univerzitet u Beogradu (Serbia)
- Gimnazija Ivanjica (Serbia)

Rok publikace 2024

## Obsah

<b>1. PŘEDMLUVA.....</b>	<b>5</b>
<b>2. ÚVOD DO VÝVOJOVÉHO PROSTŘEDÍ GREENFOOT.....</b>	<b>7</b>
2.1. TVOŘIVÉ ZKOUMÁNÍ VÝVOJE HER .....	7
2.1.1. Příručka učitele na přípravu vyučovací hodiny .....	9
<b>3. DEFINICE TŘÍDY .....</b>	<b>12</b>
3.1. ZKOUMÁNÍ TŘÍD A OBJEKTŮ PROSTŘEDNICTVÍM VÝVOJE HER V PROSTŘEDÍ GREENFOOT .....	12
3.1.1. Příručka učitele na přípravu vyučovací hodiny .....	13
3.2. VYTVÁŘENÍ TŘÍD A OBJEKTŮ PROSTŘEDNICTVÍM VÝVOJE HER S PROSTŘEDÍM GREENFOOT.....	16
3.2.1. Příručka učitele na přípravu vyučovací hodiny .....	17
<b>4. ALGORITMUS.....</b>	<b>20</b>
4.1. ÚVOD DO ALGORITMŮ V PROSTŘEDÍ GREENFOOT .....	20
4.1.1. Příručka učitele na přípravu vyučovací hodiny .....	21
4.2. DOBRODRUŽSTVÍ V PROSTŘEDÍ GREENFOOT: VOLÁNÍ METOD V JAVĚ, PRÁCE S DOKUMENTACÍ A ŘÍZENÍ APLIKACE .....	23
4.2.1. Příručka učitele na přípravu vyučovací hodiny .....	25
<b>5. VĚTVENÍ .....</b>	<b>29</b>
5.1. ZKOUMÁNÍ VĚTVENÍ PŘI VÝVOJI HER V PROSTŘEDÍ GREENFOOT - NEÚPLNÉ VĚTVENÍ KÓDU .....	29
5.1.1. Příručka učitele na přípravu vyučovací hodiny .....	30
5.2. ZKOUMÁNÍ VĚTVENÍ PŘI VÝVOJI HER V PROSTŘEDÍ GREENFOOT - ÚPLNÉ VĚTVENÍ KÓDU.....	32
5.2.1. Příručka učitele na přípravu vyučovací hodiny .....	33
<b>6. PROMĚNNÉ A VÝRAZY .....</b>	<b>36</b>
6.1. ÚVOD DO PROMĚNNÝCH A DATOVÝCH TYPŮ V PROSTŘEDÍ GREENFOOT .....	36
6.1.1. Příručka učitele na přípravu vyučovací hodiny .....	37
6.2. ÚVOD DO OPERÁTORŮ A VÝRAZŮ V PROSTŘEDÍ GREENFOOT .....	39
6.2.1. Příručka učitele na přípravu vyučovací hodiny .....	40
6.3. ÚVOD DO KONSTRUKTORŮ V PROSTŘEDÍ GREENFOOT.....	43
6.3.1. Příručka učitele na přípravu vyučovací hodiny .....	44
6.4. ÚVOD DO ATRIBUTŮ V PROSTŘEDÍ GREENFOOT .....	45
6.4.1. Příručka učitele na přípravu vyučovací hodiny .....	46
6.5. ÚVOD DO PŘETĚŽOVÁNÍ KONSTRUKTORŮ V PROSTŘEDÍ GREENFOOT .....	48
6.5.1. Příručka učitele na přípravu vyučovací hodiny .....	49
<b>7. ASOCIACE .....</b>	<b>50</b>
V RÁMCI TEMATICKÉHO CELKU ASOCIACE BYLY VYTVOŘENÉ ČTYŘI METODICKÉ MATERIÁLY.....	50
7.1. OBJEKTY V PROSTŘEDÍ GREENFOOT A JEJICH SPOLUPRÁCE: ZKOUMÁNÍ METOD A ASOCIACÍ.....	50
7.1.1. Příručka učitele na přípravu vyučovací hodiny .....	52
7.2. GREENFOOT OBJEKTY A JEJICH SPOLUPRÁCE: PROZKOUMÁNÍ ASOCIACÍ A POKROČILÝCH VOLÁNÍ METOD .....	55
7.2.1. Příručka učitele na přípravu vyučovací hodiny .....	57
7.3. GREENFOOT OBJEKTY A JEJICH SPOLUPRÁCE: VĚŽE, STŘELY A STRATEGICKÉ INTERAKCE .....	60
7.3.1. Příručka učitele na přípravu vyučovací hodiny .....	61
7.4. OBJEKTY V PROSTŘEDÍ GREENFOOT A JEJICH SPOLUPRÁCE: STŘELY, NEPŘÁTELÉ A DYNAMIKA HRY .....	66
7.4.1. Příručka učitele na přípravu vyučovací hodiny .....	67
<b>8. DĚDIČNOST.....</b>	<b>70</b>
8.1. ÚVOD DO DĚDIČNOSTI V PROSTŘEDÍ GREENFOOT .....	71
8.1.1. Příručka učitele na přípravu vyučovací hodiny .....	72
8.2. KONCEPTY DĚDIČNOSTI V PROSTŘEDÍ GREENFOOT (ČÁST 1).....	74

8.2.1.	<i>Příručka učitele na přípravu vyučovací hodiny</i> .....	75
8.3.	KONCEPTY DĚDIČNOSTI V PROSTŘEDÍ GREENFOOT (ČÁST 2) .....	77
8.3.1.	<i>Příručka učitele na přípravu vyučovací hodiny</i> .....	79
8.4.	KONCEPTY DĚDIČNOSTI V PROSTŘEDÍ GREENFOOT (ČÁST 3) .....	80
8.4.1.	<i>Příručka učitele na přípravu vyučovací hodiny</i> .....	81
<b>9.</b>	<b>ZAPOUZDŘENÍ</b> .....	<b>82</b>
9.1.	ZKOUMÁNÍ ZAPOUZDŘENÍ PROSTŘEDNICTVÍM VÝVOJE HER V PROSTŘEDÍ GREENFOOT (ČÁST 1) .....	82
9.1.1.	<i>Příručka učitele na přípravu vyučovací hodiny</i> .....	84
9.2.	ZKOUMÁNÍ ZAPOUZDŘENÍ PROSTŘEDNICTVÍM VÝVOJE HER V PROSTŘEDÍ GREENFOOT (ČÁST 2) .....	86
9.2.1.	<i>Příručka učitele na přípravu vyučovací hodiny</i> .....	87

## Seznam tabulek

Tabulka 1.	Šablona na dokumentování metodických materiálů .....	6
Tabulka 2.	Tvořivé zkoumání vývoje her .....	7
Tabulka 3.	Zkoumání tříd a objektů prostřednictvím vývoje her v prostředí Greenfoot .....	12
Tabulka 4.	Vytváření tříd a objektů prostřednictvím vývoje her s prostředím Greenfoot .....	16
Tabulka 5.	Úvod do algoritmů v prostředí Greenfoot .....	20
Tabulka 6.	Dobrodružství v prostředí Greenfoot: Volání metod v Javě, práce s dokumentací a řízení aplikace .....	23
Tabulka 7.	Zkoumání větvení při vývoji her v prostředí Greenfoot - Neúplné větvení kódu .....	29
Tabulka 8.	Zkoumání větvení při vývoji her v prostředí Greenfoot - Úplné větvení kódu .....	32
Tabulka 9.	Úvod do proměnných a datových typů v prostředí Greenfoot .....	36
Tabulka 10.	Úvod do operátorů a výrazů v prostředí Greenfoot .....	39
Tabulka 11.	Úvod do konstruktorů v prostředí Greenfoot .....	43
Tabulka 12.	Úvod do atributů v prostředí Greenfoot .....	45
Tabulka 13.	Úvod do přetěžování konstruktorů v prostředí Greenfoot .....	48
Tabulka 14.	Objekty v prostředí Greenfoot a jejich spolupráce: zkoumání metod a asociací .....	50
Tabulka 15.	Greenfoot objekty a jejich spolupráce: prozkoumání asociací a pokročilých volání metod .....	55
Tabulka 16.	Greenfoot objekty a jejich spolupráce: Věže, střely a strategické interakce .....	60
Tabulka 17.	Objekty v prostředí Greenfoot a jejich spolupráce: střely, nepřátelé a dynamika hry .....	66
Tabulka 18.	Úvod do dědičnosti v prostředí Greenfoot .....	71
Tabulka 19.	Koncepty dědičnosti v prostředí Greenfoot (část 1) .....	74
Tabulka 20.	Koncepty dědičnosti v prostředí Greenfoot (část 2) .....	77
Tabulka 21.	Koncepty dědičnosti v prostředí Greenfoot (část 3) .....	80
Tabulka 22.	Zkoumání zapouzdření prostřednictvím vývoje her v prostředí Greenfoot (část 1) .....	82
Tabulka 23.	Zkoumání zapouzdření prostřednictvím vývoje her v prostředí Greenfoot (část 2) .....	86

# 1. Předmluva

Cílem této knihy je představit kurz s materiály, které pomohou učitelům při přípravě materiálů na vyučování v oblasti řešení programovacích úloh s využitím základů objektově-orientovaného programování (OOP) podle principů light OOP („zjednodušeného OOP“).

Studenti se naučí rozdělit zadané úlohy mezi spolupracující objekty, určit jejich kompetence a implementovat navržený model. Předmět nevyžaduje předchozí znalosti programování. Vyučuje se v programovacím jazyce Java. Používá vývojové prostředí Greenfoot, které podporuje programovací jazyk Java. Java je v současnosti velmi populární a v praxi široce používaný programovací jazyk. Greenfoot navíc představuje editor zdrojového kódu založený na rámcích, který využívá jazyk Stride. To otevírá možnosti pro učitele, kteří budou chtít používat v tomto učebním plánu prezentované techniky se studenty mladšího věku. Greenfoot je velmi vizuální a od začátku umožňuje vytvořit vizualizovaný objekt, který je „živý“ a se kterým se dají provádět interakce. Proto je teoretický úvod minimalizovaný a studenti začnou pracovat hned od začátku.

Kurz vysvětluje jednoduché koncepty OOP (jako je zapouzdření, dědičnost nebo asociace) na tvorbě počítačových her, kde dochází k jednoduchému a intuitivnímu vnímání těchto konceptů. Proces tvorby počítačové hry je založený na týmové práci a prakticky využívá vědomosti a dovednosti z jiných oblastí informatiky a s předměty, které s ní souvisí (práce s multimédií a kancelářským softwarem). Design každé počítačové hry je dostatečně flexibilní na to, aby studenti mohli hru individuálně a tvořivě rozšířit. Kromě toho vede návrh ke správnému využití získaných vědomostí.

Kniha je zaměřená na představení inovativního přístupu k výuce programování, založeného na řešení úloh s využitím paradigmatu objektově-orientovaného programování (OOP). OOP je v současnosti dominantním paradigmatem pro vývoj aplikací. Proto je vhodné, aby studenti měli vědomosti a dovednosti z této oblasti. Předmět představuje vývojové prostředí, které využívá různé formy úpravy zdrojového kódu (buď rámcová úprava pomocí zjednodušené formy, nebo i reálný zápis zdrojového kódu), což umožňuje vyučovat studenty na různých úrovních předchozích technických znalostí a dovedností. Svou jednoduchostí a přehledností tento nástroj podporuje rychlé a intuitivní pochopení vyučovaných témat, což má pozitivní vliv na studenty a jejich motivaci.

Prostřednictvím programování interaktivních her v grafickém prostředí student získá vědomosti a dovednosti, takže bude schopný:

- identifikovat problém,
- identifikovat vhodné objekty na řešení identifikovaného problému (dekompozice objektů),
- navrhnout třídy objektů, stejně tak jako jejich atributy a metody,
- identifikovat a správně využívat vztahy mezi objekty (asociace, dědičnost),
- navrhnout algoritmus na řešení problému a rozdělit ho mezi spolupracující objekty,
- používat prvky zdrojového kódu (větvení, cykly) na implementaci navrženého algoritmu,
- efektivně používat prostředky na debugování (ladění) zdrojového kódu,
- vytvořit jednoduchou aplikaci s grafickým rozhraním ve vývojovém prostředí Greenfoot.

Výsledky vzdělávání z předmětu jsou shrnuty v následujících bodech:

- pochopení základních principů objektově orientovaného programování,
- pochopení základů algoritmizace,
- pochopení syntaxe programovacího jazyka Java,
- analýza běhu programu na základě zdrojového kódu,
- schopnost vytvářet vlastní programy s využitím OOP.

Moderním přístupem pro koncepci vyučovacích hodin, zejména těch pro základní a střední školy, je definování a sdílení výukových scénářů. Metodické materiály (MM) „jsou vnímány jako moderní pedagogický přístup, který umožňuje individualizaci vyučovacího procesu s ohledem na různé potřeby studentů. Vyučování založené na MM se zaměřuje na relevantní vědomosti a dovednosti studentů včetně těch, které jsou potřebné pro digitální společnost. Pečlivé plánování MM může odstranit možná úskalí a nedostatky, které by mohli ovlivnit vyučovací proces.“

V kontextu vzdělávání a návrhu výukových scénářů představují MM podrobné popisy nebo popisy, které načrtávají konkrétní vyučovací situaci nebo kontext. Tyto MM se často používají při vzdělávání učitelů na simulaci reálných vyučovacích situací, a proto je považujeme za nejlepší nástroj na prezentaci našich inovativních nápadů v oblasti vyučování a učení se. Vzhledem k tomu, že metodické materiály obvykle obsahují informace o cílech vyučování, obsahu, který se má být vyučován, charakteristikách studentů, použitých vyučovacích metodách a strategiích hodnocení, lze je sladit i s prvky potřebnými pro naše vzdělávací tematické plány.

Abychom měli strukturovaný přístup při definování několika MM, definovali jsme následující šablonu, která se naplní konkrétními informacemi, které se týkají konkrétních témat. Šablona obsahuje krátký popis toho, jakým způsobem definovat jednotlivé prvky MM.

Tabulka 1. Šablona na dokumentování metodických materiálů

<i>Název</i>	Přidat popisný a poutavý název.
<i>Cíle vzdělávání</i>	Jasně uvést plánované výsledky vzdělávání. Co by měli studenti vědět, čemu by měli rozumět nebo co by měli být schopni vytvořit po zvládnutí metodiky
<i>Cílová skupina</i>	Uvést cílovou skupinu, ročník a již existující vědomosti, pro které je metodický materiál (MM) určený.
<i>Trvání scénáře</i>	Odhadnout čas potřebný na dokončení MM včetně všech konkrétních časových rámců pro různé aktivity. Např. motivace (5 min), zkoumání, které vykonají studenti samostatně (10 min), programování řešení v týmu (20 min), prezentace/diskuse (10 min.).
<i>Materiály a zdroje</i>	Uvést seznam materiálů, zdrojů a nástrojů potřebných pro učitele a studenty. To může zahrnovat učebnice, online materiály a multimediální zdroje, software atd.
<i>Popis</i>	<ul style="list-style-type: none"> <li>• Představit cíl vyučování, vysvětlit jeho účel a význam.</li> <li>• Načrtnout základní aktivity/úlohy, které budou studenti vykonávat k dosažení cílů vzdělávání. Uvést podrobnosti, jako je diskuse, praktická činnost, skupinová práce, soutěž atd.</li> <li>• Uvést, jakým způsobem budou žáci organizováni, tj. zda budou pracovat individuálně nebo v týmu. Jak velké budou týmy?</li> <li>• Vysvětlit, na jakých projektech/problémech/úlohách budou studenti pracovat. Doporučit použití problémového nebo projektového přístupu. Ty by měli odrážet situace z reálného života.</li> </ul>

	<ul style="list-style-type: none"> <li>• Vysvětlit, jakým způsobem budou projekty/problémy/úlohy přidělené studentům (týmům).</li> <li>• V případě týmové práce uvést podrobnosti o tom, jakým způsobem budou studenti spolupracovat.</li> <li>• Uvést víc podrobností týkajících se aktivit, kterých se studenti musí zúčastnit.</li> <li>• Je-li použita obrácená třída, uvést, kterou část daného tématu musí studenti sami prozkoumat.</li> </ul>
<i>Hodnocení</i>	<p>Uvést podrobnosti týkající se toho, jakým způsobem se bude hodnotit úsilí a vědomosti studentů.</p> <ul style="list-style-type: none"> <li>• kdo bude studenty hodnotit: (1) učitel, (2) studenti ohodnotí svoji vlastní práci (sebehodnocení), (3) studenti si navzájem ohodnotí svoji práci (vzájemné hodnocení).</li> <li>• která kritéria hodnocení budou použita?</li> <li>• jak často se bude hodnotit?</li> <li>• atd.</li> </ul>
<i>Šíření výsledků</i>	<p>Vysvětlit, jakým způsobem budou studenti šířit své výsledky (přes učitele, přes spolužáky). Např. studenti (všichni nebo podskupina) mohou prezentovat své výsledky/řešení před učitelem a svými spolužáky, a potom může následovat porovnání a diskuse.</p>

V souladu s tím najdete náš tematický plán na této webové adrese: Podrobnosti o kurzu ([learning-design.eu](http://learning-design.eu)). Materiály, které podporují tuto knihu, najdete na platformě Moodle na adrese: <https://oop4fun.fon.bg.ac.rs/>.

## 2. Úvod do vývojového prostředí Greenfoot

Greenfoot je vizuální 2D vzdělávací softwarový nástroj s editorem programu na vytváření her a simulací v programovacím jazyce Java. Greenfoot je vizuální a interaktivní. Programování probíhá ve standardním textovém programovacím jazyce Java, což poskytuje kombinaci zkušeností s programováním v tradičním textovém jazyce s vizuálním prováděním.

### 2.1. Tvořivé zkoumání vývoje her

Tabulka 2. Tvořivé zkoumání vývoje her

<i>Název</i>	Tvořivé zkoumání vývoje her
<i>Cíle vzdělávání</i>	Na konci této kapitoly budou studenti mít nejen úspěšně nainstalovaný Greenfoot a ověřené jeho možnosti na ukázkovém projektu, ale zapojí se také do společné praktické hry ve vývojovém prostředí. Tento hravý úvod udává tón bezprostřednímu zkoumání vývoje her, podporuje tvořivost, týmovou práci a nadšený přístup k programování v prostředí Greenfoot.

<i>Cílová skupina</i>	Studenti středních škol na kurzu OOP4Fun Základní znalosti programování včetně proměnných, funkcí, iterací a konceptů výběru.
<i>Trvání scénáře</i>	<ol style="list-style-type: none"> <li>1. Úvod (5 min) .....</li> <li>2. Rychlá výzva (10 min) .....</li> <li>3. Hraní her s učitelem (30 min) .....</li> <li>4. Vytvoření týmu a zadání projektu (5 min) .....</li> <li>5. Týmová spolupráce a programování (30 min) .....</li> <li>6. Vzájemné hodnocení a zpětná vazba (10 min) .....</li> <li>7. Domácí úkol (30 min).....</li> <li>8. Klasifikace soutěže (30 min) .....</li> </ol>
<i>Materiály a zdroje</i>	<ul style="list-style-type: none"> <li>• Webová stránka Greenfoot a pokyny ke stažení.</li> <li>• Příklady připravené učitelem.</li> <li>• Internetové zdroje na identifikaci dalších příkladů.</li> </ul>
<i>Popis</i>	<p>V tomto 90-minutovém výukovém scénáři se studenti středních škol ponoří do světa prostředí Greenfoot prostřednictvím hry, zábavy, výzkumu, bádání a týmové práce.</p> <p>Poté, co učitel představí dnešní lekci a stanoví cíle, začne rychlá výzva. Studenti dostanou gamifikovanou úlohu najít návod, stáhnout na nainstalovat program Greenfoot (to je pro ně zatím neznámý vývojový nástroj) do svých počítačů. První tři studenti dostanou ocenění (odznaky, body, skóre, sladkosti atd.).</p> <p>Druhým překvapením pro ně je, že v následujících 30 minutách budou hrát hry s učitelem. Jedná o střetnutí pod vedením učitele zaměřené na otevření, kompilaci a spuštění jednoho-dvou jednoduchých ukázkových projektů (na lehké až střední úrovni složitosti). Tím se studenti seznámí se základními prvky vývojového prostředí Greenfoot a se základními postupy při práci se soubory a vlastnostmi projektu.</p> <p>Potom budou studenti rozděleni do týmů (po 3 - 4 studentech) a dostanou jednoduché zadání. Týmy by měly „něco“ v daném ukázkovém projektu změnit, aby hra byla překvapivá nebo zábavná. V rámci týmové spolupráce a programování (30 minut) budou týmy společně pracovat na tom, aby se pokusily něco v daných příkladech změnit. V situaci, že poruší program za hranicí toho, co dokáží sami opravit, mohou požádat o pomoc učitele případně si mohou znova stáhnout „startovací verzi“ projektu. Toto bude dobrý příklad, proč by se měl při programování používat verzovací systém</p> <p>Jeden nebo dva týmy představí svoji práci na vzájemné posouzení, zpětnou vazbu a prodiskutují výsledky s učitelem.</p> <p>V rámci domácí úlohy by měl každý student vyhledat příklady her v prostředí Greenfoot a měl by své třídě představit svůj oblíbený příklad tak, že nahraje internetový odkaz a popis toho, proč je ten konkrétní příklad jeho oblíbeným. Měl také poskytnout ukázkou dvou až tří screenshotů vývojového prostředí a spuštěné hry. V rámci gamifikace a motivace prostřednictvím soutěže by měl každý student hlasovat pro tři nejlepší hry (není dovoleno hlasovat za vlastní hru). Vítězové budou vyhlášeni a odměněni žetony uznání (odznaky, body, skóre, sladkosti atd.).</p>
<i>Hodnocení</i>	<p>Tato aktivita umožní učitelům poskytovat formativní zpětnou vazbu na základě diskuse a monitorování obrácené třídy a týmové práce studentů.</p> <p>Gamifikace představuje nejen neformální hodnocení, ale také zvýší zájem, vnitřní motivaci a výstupy vzdělávání celé skupiny. Hodnocení vzájemného hodnocení se bude provádět</p>



	online jako součást domácí úlohy. To studentům připomene důležité aspekty hodiny, přinutí je nainstalovat Greenfoot, uvede různé příklady a připomene jim to, co se v průběhu vyučování dělalo, což zvýší dosáhnuté výsledky vzdělávání celé skupiny.
<i>Šíření výsledků</i>	Na šíření výsledků mezi učiteli a studenty se použije repozitář Github/Gitlab a systém řízení výuky (např. Moodle). Studenti mohou pokračovat v diskusi na dané téma na fóru, které je jim poskytnuté prostřednictvím nástroje na řízení výuky.

### 2.1.1. Příručka učitele na přípravu vyučovací hodiny

#### 1. Úvod (5 min)

##### Cíl:

Studenti se seznámí s programovacími jazyky (vizuálními a textovými), integrovanými vývojovými prostředími a zdrojovým kódem.

##### Pojmy na diskusi:

- programovací jazyky, integrované vývojové prostředí a zdrojový kód

##### Aktivita:

Učitel udělá krátký úvod k dnešní hodině. Učitel představí pojmy jako:

- programovací jazyk
- integrované vývojové prostředí (programovací nástroje)
- zdrojový kód.

Učitel prezentuje příklady zdrojových kódů v různých programovacích jazycích, jako jsou Java, C a Python. Dále předvádí i programy v jazyce Scratch nebo jiném vizuálním jazyce. Učitel nejde "do hloubky", ale snaží se tyto pojmy vysvětlit na jednoduchém příkladě, například se snaží porovnat učení se programovacího jazyka s učením se rodného nebo cizího jazyka.

- Přirozené jazyky mají svoji vlastní gramatiku a pravopis, stejně je to i v případě programovacích jazyků. Při psaní se řídíme určitými gramatickými a pravopisnými pravidly, podobně je tomu tak i při zápisu programu v programovacím jazyce
- Když píšeme příběh v přirozeném jazyce, používáme sešit, pero jako nástroj, který nám při tom pomáhá, podobně když zapisujeme program v programovacím jazyce je našim "nástrojem" integrované vývojové prostředí.
- Výsledkem psaní může být příběh napsaný na papíře, zatímco výsledkem programování je program, který vytvoříme a který označujeme jako zdrojový kód.

Učitel může předvést úvod na vlastním příkladu.

#### 2. Rychlá výzva (10 min)

##### Cíl:

Studenti prozkoumají Greenfoot a návod na jeho instalaci.

##### Pojmy na diskusi:

- Greenfoot, návod na instalaci

##### Aktivita:

Učitel zadá studentům úlohu zjistit, co je to Greenfoot. Učitel poví studentům, aby našli návod na instalaci Greenfootu. Studenti pracují na úloze, potom jim učitel představí, jak najít, stáhnout a spustit instalaci Greenfootu. Návod stažení a instalaci Greenfootu může učitel poskytnout přes platformu Moodle.

### 3. Hraní her s učitelem (30 min)

#### Cíl:

Studenti spustí různé projekty Greenfoot.

#### Pojmy na diskusi:

- Greenfoot projekt (webové a samostatné projekty)

#### Aktivita:

Učitel zadá studentům úlohu, aby si na se podívali po internetu našli příklady projektů vytvořených v prostředí Greenfoot. Studenti hledají projekty, zatímco učitel sleduje jejich práci.

Učitel představí studentům, jakým způsobem mohou najít příklady hotových projektů vytvořených v prostředí Greenfoot. Například ve vyhledávači Google je možné projekty najít na základě klíčových slov "Greenfoot Java project example" nebo podobně.

Učitel vysvětlí studentům, že existují dva typy projektů v prostředí Greenfoot

- jeden, který možné spustit ve webovém prohlížeči
- další, který je možné stáhnout, potom otevřít a spustit v prostředí Greenfoot.

Učitel představí projekty:

- projekty, které možné spustit přímo ve webovém prohlížeči,
- projekty, které možné spustit v prostředí Greenfoot po stažení.

Učitel si může stáhnout několik ukázkových projektů, nebo se k nim dostat pomocí odkazů.

### 4. Vytvoření týmu a zadání projektu (5 min)

#### Cíl:

Student se zapojí do projektového vyučování pomocí jednoduchého zadání.

#### Pojmy na diskusi:

#### Aktivita:

Učitel vytvoří týmy, připraví úlohu a zadá projekt, na kterém budou studenti pracovat. Některé příklady úloh najdete na Moodlu. Učitel vybere jeden reprezentativní projekt (ne moc složitý) a pro tento projekt definuje úlohu (problém), kterou mají studenti vyřešit.

### 5. Týmová spolupráce a programování (30 min)

#### Cíl:

Student umí vytvořit projekt v prostředí Greenfoot. Studenti pracují na zadání úlohy.

#### Pojmy na diskusi:

- Greenfoot projekt

#### Aktivita:

Učitel spustí Greenfoot a představí studentům, jaký způsobem vytvořit projekt. Učitel zadá studentům úlohu vytvořit projekt v prostředí Greenfoot.

**Úloha 1.1:** Vytvořte nový projekt, dejte mu správný název (např. TowerDefense) a uložte ho na správné místo.

Učitel zdůrazní studentům, že projekt (totožný s projektem, který byl právě vytvořen) může být:

- stažen z git repozitáře z následujícího odkazu  
**Commit:** [9046f5353d857dcc112abd92d7b7170abcc64a80](https://github.com/9046f5353d857dcc112abd92d7b7170abcc64a80)
- stažen jako zip soubor z následujícího odkazu  
<https://oop4fun.fon.bg.ac.rs/>

Učitel zdůrazní studentům, že na dnešní hodině nebudou pracovat na projektu, který právě vytvořili, ale až od následující hodiny, a že nyní budou pracovat na existujících projektech.

Učitel vyzve studenty, ať zkusí v projektu něco pozměnit a ovlivnit tím chování hry. Studenti se chopí úlohy a řeší problém. Učitel sdělí studentům, že v případě poškození kódu mohou projekt stáhnout a otevřít znova, případně ho požádat o pomoc.

## 6. Vzájemné hodnocení a zpětná vazba (10 min)

**Cíl:**

Studenti umí diskutovat o jejich navrženém řešení.

**Pojmy na diskusi:**

**Aktivita:**

Učitel sleduje práci studentů a v případě potřeby jim dává pokyny (instrukce). Po skončení učitel vybere jeden tým, který ukáže své řešení. Učitel diskutuje se studenty o navrhovaném řešení.

## 7. Domácí úkol (30 min)

**Cíl:**

Studenti zvládnou prozkoumat projekty v prostředí Greenfoot.

**Pojmy na diskusi:**

**Aktivita:**

Učitel definuje úlohu, kterou mají studenti vyřešit na jednom z existujících projektů.

## 8. Klasifikace soutěže (30 min)

**Cíl:**

Studenti se zapojí do procesu hodnocení jejich projektů.

**Pojmy na diskusi:**

**Aktivita:**

Studenti prezentují svůj projekt. Učitel vyžaduje, aby se všichni zúčastnili na hodnocení prezentovaných projektů například zasláním odkazů ostatním studentům na vyplnění formuláře Google. Studenti vyhodnotí tři nejlepší týmy.

Po prezentaci provede učitel souhrn studentských projektů. Učitel se podělí o své dojmy z jejich práce a uvede, zda je spokojený, zda práce splnila jeho očekávání, nebo zda je překonala.

### 3. Definice třídy

V rámci tematického celku Definice třídy byly vytvořeny dva metodické materiály.

#### 3.1. Zkoumání tříd a objektů prostřednictvím vývoje her v prostředí Greenfoot

Tabulka 3. Zkoumání tříd a objektů prostřednictvím vývoje her v prostředí Greenfoot

<i>Název</i>	Zkoumání tříd a objektů prostřednictvím vývoje her v prostředí Greenfoot
<i>Cíle vzdělávání</i>	Na konci této kapitoly budou studenti schopni pochopit základní koncept <b>objektu a třídy</b> . Pochopení zkoumaných konceptů se bude diskutovat v kontextu vývoje hry, což podpoří tvořivost, týmovou práci a motivující přístup k programování s nástrojem Greenfoot.
<i>Cílová skupina</i>	Studenti středních škol na kurzu OOP4Fun. Základní znalosti programování včetně objektů a tříd.
<i>Trvání scénáře</i>	<ol style="list-style-type: none"> <li>1. Objekt (10 min).....</li> <li>2. Identifikace objektů a jejich vlastností (15 min).....</li> <li>3. Třída, instance, dědičnost (15 min) .....</li> <li>4. Orientace v prostředí Greenfoot (10 min).....</li> <li>5. Konstruktor třídy a úloha 1.2 (15 min) .....</li> <li>6. Nastavení obrázku (10 min).....</li> <li>7. Příprava grafiky světa – úloha 1.3 (15 min) .....</li> </ol>
<i>Materiály a zdroje</i>	<ul style="list-style-type: none"> <li>• Webová stránka Greenfoot a pokyny na stažení.</li> <li>• Příklady připravené učitelem.</li> <li>• Internetové zdroje jako ukázky dalších příkladů.</li> </ul>
<i>Popis</i>	<p>V tomto 90-minutovém výukovém scénáři se studenti středních škol seznámí s principy programování související s pojmy <b>objekt a třída</b> z pohledu vývoje her pomocí nástroje Greenfoot.</p> <p>Hodina začíná 10-minutovým úvodem učitele, který uvede studenty do světa objektově-orientovaného programování tím, že vysvětlí pojem objektu a jeho vlastností v reálném životě.</p> <p>Potom učitel zadá studentům úlohu v rychlé výzvě (10 minut), v které musí na základě textového popisu úlohy identifikovat předměty a jejich vlastnosti. Potom učitel spolu se studenty vymyslí řešení úlohy (5 minut).</p> <p>V pokračování hodiny učitel vysvětlí rozdíl mezi třídou a objektem. Na nejvyšší úrovni abstrakce vysvětlí pojem dědičnost (15 minut).</p> <p>Učitel spustí prostředí Greenfoot a vysvětlí třídy <b>World</b>, <b>Actor</b> a <b>MyWorld</b> (10 minut).</p> <p>Učitel vysvětlí a prezentuje zdrojový kód, který byl vygenerovaný a který je uvnitř tříd <b>World</b>, <b>Actor</b> a <b>MyWorld</b> po vytvoření projektu (15 minut).</p>

	<p>Učitel spustí úlohu 1.2 a ukáže studentům jakým způsobem nastavit obrázek pro konkrétní třídu (10 minut).</p> <p>Učitel vysvětlí, jakým způsobem vytvořit svět pro aplikaci, kterou potřebují vytvořit a pracuje s nimi na úloze 1.3 (15 minut).</p>
<i>Hodnocení</i>	Gamifikace představuje nejen neformální hodnocení, ale také zvýší zájem, vnitřní motivaci a výsledky vzdělávání celé skupiny.
<i>Šíření výsledků</i>	Na šíření výsledků mezi učiteli a studenty se použije repozitář Github/Gitlab a systém řízení výuky (např. Moodle). Studenti mohou pokračovat v diskusi na dané téma na fóru, které je jim poskytnuté prostřednictvím nástroje na řízení výuky.

### 3.1.1. Příručka učitele na přípravu vyučovací hodiny

#### 1. Objekt (10 min)

**Cíl:**

Studenti se seznámí s pojmem objekt na příkladech ze života.

**Pojmy na diskusi:**

- objekty a jejich vlastnosti

**Aktivita:**

Učitel zavede pojem objekt. Učitel by měl tento pojem přiblížit studentům pomocí příkladů z reálného života. Například by se mohl studentů zeptat na jejich jméno, výšku, datum narození, barvu očí atd. Učitel klade tyto otázky, aby donutil studenty zamyslet se nad tím, čím se od sebe liší a připravil studenty tak na další otázku: Jakým způsobem se studenti od sebe navzájem odlišují?

Učitel konstatuje, že každý z nás (učitel, student) má vlastnosti, kterými se od sebe vzájemně odlišujeme, a zdůrazňuje, že každý z nás je vlastně "jeden" objekt. Učitel vysvětlí pojem vlastnost jako vlastnost, kterou má každý z nás, spolu s konkrétní hodnotou každé vlastnosti, která je jedinečná pro každého jednotlivce. Objekty se tedy od sebe navzájem odlišují na základě hodnot, které mají pro své příslušné vlastnosti.

#### 2. Identifikace objektů a jejich vlastností (15 min)

**Cíl:**

Studenti zvládají identifikovat předměty a jejich vlastnosti.

**Pojmy na diskusi:**

- objekty a jejich vlastnosti

**Aktivita:**

Učitel zadá studentům úlohu, aby na základě textu určili předměty a jejich vlastnosti.

### 3. Třída, instance, dědičnost (15 min)

#### Cíl:

Studenti získají vědomosti o třídě a instanci. Umějí rozlišovat mezi třídou a instancí.

#### Pojmy na diskusi:

třída, instance

#### Aktivita:

Na vysvětlení pojmu třída, instance třídy (objekt) a dědičnost učitel uvádí příklady z reálného života. Učitel klade studentům otázky, aby pochopili rozdíl mezi třídou a objektem. Učitel vede diskusi o rozpoznávaných objektech a jejich zařazení do tříd.

### 4. Orientace v prostředí Greenfoot (10 min)

#### Cíl:

Studenti zvládnou úvod do světa v prostředí Greenfoot.

#### Pojmy na diskusi:

- instance světa v prostředí Greenfoot

#### Aktivita:

Učitel spustí prostředí Greenfoot a vytvoří v něm jednoduchý projekt. Učitel použije tento projekt na vysvětlení, prostředí Greenfoot.

Učitel upozorní že každý projekt vytvořený v prostředí Greenfoot obsahuje 3 třídy: **World**, **Actor** a **MyWorld**.

Učitel představí třídu **MyWorld** a její úlohu.

Učitel zdůrazní, že pozadí každé aplikace vytvořené v prostředí Greenfoot se skládá z buněk, které představují jednu matici. Učitel ukáže, jakým způsobem definovat velikost pozadí (rozměr matice) a velikost každé buňky matice.

Učitel vysvětlí, že objekty, které se zobrazují v tabulce („scéna“), se nachází na jedné z těchto buněk. Učitel ukáže zdrojový kód každé třídy (**World**, **Actor** a **MyWorld**).

### 5. Konstruktor třídy a úloha 1.2 (15 min)

#### Cíl:

Studenti se zapojí do práce na projektové úloze a pochopí konstruktor světa.

#### Pojmy na diskusi:

- svět v prostředí Greenfoot, konstruktor světa

#### Aktivita:

Učitel prezentuje zdrojový kód a představí pojem konstruktor světa.

Učitel zadá studentům úlohu 1.2: Vytvořte svět s velikostí 24x12 buněk. Každá buňka by měla mít velikost 50x50 bodů.

Učitel sleduje práci studentů a na závěr požádá jednoho z nich, aby ukázal své řešení a popsal ho.

**Commit:** [a593cd4a92d0fa0db78275614c3e41a2e96b4e57](https://commit.coursera.org/a593cd4a92d0fa0db78275614c3e41a2e96b4e57)

## 6. Nastavení obrázku (10 min)

### Cíl:

Nastavení obrázku světa v projektu Greenfoot.

### Pojmy na diskusi:

- obrázek pro třídu světa

### Aktivita:

Učitel spustí aplikaci po dokončení úlohy 1.2. Učitel vysvětlí studentům, že pozadím aplikace v prostředí Greenfoot (resp. světa) může být obrázek. Učitel studentům vysvětlí, že pozadí může být buď jeden obrázek pokrývající celou plochu světa, nebo obrázek s velikostí, která odpovídá rozměrům buňky.

Učitel ukáže, jakým způsobem nastavit obrázek pro třídu **MyWorld**. Studenti postupují podle pokynů učitele a pracují společně krok za krokem.

## 7. Příprava grafiky světa – úloha 1.3 (15 min)

### Cíl:

Studenti se seznámí s možnostmi, jak nastavit pozadí třídy World.

### Pojmy na diskusi:

- pozadí třídy World

### Aktivita:

Učitel zadá studentům úlohu 1.3: Vytvořte vhodný obrázek pozadí pro třídu **World** a nastavte ho. Učitel vysvětlí studentům, co od nich očekává. Učitel může stáhnout závěrečný projekt, který už je dopředu připravený a ukáže jim ho. Učitel poskytne studentům odkaz na stažení původního projektu, který je třeba aktualizovat pomocí této úlohy. Původní projekt si lze stihnout např. z úložiště git.

Studenti na úloze pracují samostatně nebo ve skupině. Učitel monitoruje práci studentů.

Učitel demonstruje řešení krok za krokem, zatímco studenti postupují podle jeho pokynů.

**Popis řešení:** studenti se pokusí o vytvoření řešení:

- Najděte nebo vytvořte vhodný obrázek pro pozadí světa. Můžete použít buď připravené obrázky (vyberte položku „**Nastavit obrázek**“ z kontextového menu třídy MyWorld), anebo vlastní obrázek (zkopírujte obrázek do podsložky **images** ve složce projektu a vyberte ho stejným způsobem).
- Jako pozadí můžete použít jediný obrázek, který pokryje celou plochu světa (vypočítejte potřebnou velikost obrázku vzhledem k velikosti světa) nebo menší obrázek, který se bude opakovaně kopírovat (použijte čtvercový obrázek o velikosti buňky).

**Commit:** [1184980643db082cfdd6bde9984bceaddf010d49](https://commit.coursera.org/1184980643db082cfdd6bde9984bceaddf010d49)

### 3.2. Vytváření tříd a objektů prostřednictvím vývoje her s prostředím Greenfoot

Tabulka 4. Vytváření tříd a objektů prostřednictvím vývoje her s prostředím Greenfoot

<b>Název</b>	Vytváření tříd a objektů prostřednictvím vývoje her s prostředím Greenfoot
<b>Cíle vzdělávání</b>	Na konci této kapitoly studenti pochopí základní pojmy objektu, třídy, vlastnosti a metody třídy.
<b>Cílová skupina</b>	Studenti středních škol, kteří na kurzu OOP4Fun. Základní znalosti programování včetně objektů, tříd, vlastností tříd a metod.
<b>Trvání scénáře</b>	<ol style="list-style-type: none"> <li>1. Základní pojmy (25 min).....</li> <li>2. Úloha 1.4 (10 min).....</li> <li>3. Úloha 1.5 (30 min).....</li> <li>4. Rozhraní objektu (5 min).....</li> <li>5. Zpráva a metoda (15 min).....</li> <li>6. Úloha 1.6 (30 min).....</li> <li>7. Opakování teorie (15 min) .....</li> </ol>
<b>Materiály a zdroje</b>	<ul style="list-style-type: none"> <li>● Webová stránka Greenfoot a pokyny na stažení.</li> <li>● Příklady připravené učitelem.</li> <li>● Internetové zdroje jako ukázky dalších příkladů.</li> </ul>
<b>Popis</b>	<p>V tomto 130-minutovém výukovém scénáři studenti projdou několika strukturovanými úlohami s cílem prohloubit své porozumění konceptů objektově orientovaného programování.</p> <p>Nejdříve stráví 25 minut vytvářením třídy s názvem <b>Enemy</b>.</p> <p>Pak bude následovat cílená 10-minutová úloha na definování atributů a metod v rámci této třídy.</p> <p>Následně se vyhradí 30 minut na vytvoření instance objektu třídy Enemy, přičemž uplatní své vědomosti o vytváření a inicializaci objektů.</p> <p>Potom se studenti v průběhu 5 minut budou věnovat koncepci rozhraní objektu s důrazem na definování toho, jaké operace může objekt vykonávat.</p> <p>Následně se studenti v průběhu 15 minut seznámí s pojmy <b>zpráva a metoda</b>, přičemž se naučí, jakým způsobem objekty komunikují prostřednictvím volání metod.</p> <p>Dalších 30 minut bude věnovaných praktické aplikaci, v rámci které studenti budou posílat zprávy instanci třídy <b>Enemy</b>, kterou vytvořili, čímž si upevní pochopení chování objektů.</p> <p>Nakonec se v rámci 15-minutové teoretické revize zopakují klíčové pojmy, jako jsou <b>objekty, třídy, instance, vnitřní stav, identita, zpráva a metody</b>, čímž se zabezpečí komplexní pochopení učiva.</p>
<b>Hodnocení</b>	Gamifikace představuje nejen neformální hodnocení, ale navíc zvýší zájem, vnitřní motivaci a



	vzdělávací výstupy celé skupiny.
<i>Šíření výsledků</i>	Na šíření výsledků mezi učiteli a studenty se použije repozitář Github/Gitlab a systém řízení výuky (např. Moodle). Studenti mohou pokračovat v diskusi na dané téma na fóru, které je jim poskytnuté prostřednictvím nástroje na řízení výuky.

### 3.2.1. Příručka učitele na přípravu vyučovací hodiny

#### 1. Základní pojmy (25 min)

##### **Cíl:**

Studenti mají znalosti o pojmu třída, o jejích vlastnostech a o jejích instancích (objektech).

##### **Pojmy na diskusi:**

- třídy, vlastnosti a objekty

##### **Aktivita:**

V úvodu hodiny učitel se studenty zopakuje pojmy, které se probíraly dříve. Prostřednictvím diskuse učitel objasní pojmy třída, vlastnosti tříd a instance tříd (objekty).

Učitel zadá studentům písemnou úlohu, ve které jim dá pokyn, aby v poskytnutém textu identifikovali konkrétní třídy, jejich vlastnosti, související objekty a příslušné hodnoty.

Učitel vybere studenta ze třídy, který představí své řešení. V průběhu této aktivity se ostatní studenti zapojí tím, že vyjadřují své názory.

Vyučující krok za krokem popíše, jakým způsobem vytvořit třídu v prostředí Greenfoot. Studenti postupují podle pokynů učitele a spolupracují krok za krokem.

#### 2. Úloha 1.4 (10 min)

##### **Cíl:**

Studen je schopen zapojit se do práce na projektu.

##### **Pojmy na diskusi:**

- vytvoření třídy

##### **Aktivita:**

Učitel zadá studentům úlohu 1.4: Vytvořte třídu Enemy a přidejte k ní vhodný obrázek.

Učitel vysvětlí studentům, co od nich očekává. Učitel může stáhnout závěrečný projekt, který dopředu připravil, a ukázat jim ho. Učitel poskytne studentům odkaz na stažení (např. přes repozitář git) původního projektu, který musí aktualizovat v rámci této úlohy.

Studenti úlohu řeší samostatně, nebo ve skupině. Učitel monitoruje práci studentů. Učitel demonstruje řešení krok za krokem, přičemž studenti postupují podle jeho pokynů. Studenti se pokusí najít řešení, které by jim pomohlo při řešení úlohy:

Popis řešení:

- Vytvořte nepřítele. Nepřítel bude pochodovat směrem k hráčově kouli, aby ji poškodil a nakonec zničil.
- Vytvořte novou podtřídu třídy **Actor** (vyberte položku „**Actor**“ kliknutím pravým tlačítkem myši a z kontextového menu třídy **Actor** vyberte položku „**Nová podtřída**“). Učitel by měl vysvětlit konvenci, že třída by měla začínat velkými písmeny, a celou konvenci názvů v programovacím jazyce Java.

Commit: [4981400623729c3d112b54454b6e6151e18426bf](https://github.com/4FUN/4FUN/commit/4981400623729c3d112b54454b6e6151e18426bf)

### 3. Úloha 1.5 (30 min)

**Cíl:**

Studenti porozumí stavu objektu. Studenti budou vědět, jak vytvořit instanci třídy v prostředí Greenfoot.

**Pojmy na diskusi:**

- stav objektu, instance

**Aktivita:**

Učitel vysvětlí pojem stav objektu na jednoduchém příkladě. Například učitel je umístěn ve třídě v určité vzdálenosti od vchodových dveří. Tato vzdálenost určuje jejich polohu, přičemž kroky vpřed nebo vzad mění jejich vzdálenost. Proto lze polohu učitele vzhledem ke vchodovým dveřím kvantifikovat pomocí proměnné, která se mění v čase. Učitelova poloha je teda definovaná jeho vzdáleností od vchodových dveří, což ilustruje, jaký je stav objektu - v tomto případě učitele - charakterizovaného hodnotou jeho atributu (vzdáleností od dveří) v daném okamžiku.

Učitel popíše úlohu 1.5, jako vytvoření instance třídy **Enemy**. Učitel otevře poslední verzi projektu, zatímco studenti postupují podle jeho pokynů a pracují společně krok za krokem.

Učitel vytvoří instanci třídy **Enemy** (vybere položku „**new Enemy ()**“ pravým tlačítkem myši z kontextového menu třídy **Enemy**, umístí ji do světa kliknutím tím, že klikne levým tlačítkem myši na požadovanou pozici). Dále se přezkoumá její vnitřní stav (vyberte se kliknutím na pravé tlačítko myši prvek umístěný do světa a z kontextového menu vytvořené instance se vyberte položka „**Prohlížet**“).

Učitel požádá studenty, aby vytvořili novou instanci a umístili ji na jinou pozici a porovnali vnitřní stavy dvou vytvořených instancí.

### 4. Rozhraní objektu (5 min)

**Cíl:**

Studenti se seznámí s rozhraním jako souborem akcí, které lze vykonat na nějakém objektu.

**Pojmy na diskusi:**

- rozhraní

**Aktivita:**

Učitel seznámí studenty s pojmem rozhraní na jednoduchých příkladech. Například, když pozorujeme člověka a činnosti, které vykonává v průběhu dne (jako je probuzení, snídaně, cesta do práce) bez toho, abychom se zaobírali konkrétními detaily - jak se budí (zda budíkem, telefonem nebo rodiči), co a kde snídá nebo jakým způsobem se přesouvá do práce - soubor těchto činností možno přirovnat k pojmu rozhraní. Rozhraní definuje, jaké činnosti mohou objekty určité třídy provádět, ale nespecifikuje, jakým způsobem se

tyto činnosti provádějí. Nedefinujeme, co si dá k snídani, jak se dostal do práce (pešky, autem nebo autobusem) nebo jak se probudil (zda ho někdo zavola, nebo zda ho probudil budík).

#### 5. Zpráva a metoda (15 min)

##### Cíl:

Studenti se seznámí s pojmem metoda.

##### Pojmy na diskusi:

- metoda třídy

##### Aktivita:

Učitel představí pojem metoda. Na vysvětlení pojmu metoda učitel spojí pojmy vlastnosti (charakteristiky, atributy) a měnící se hodnoty těchto vlastností.

Příklad:

- V případě, že se na základě údajů o vlastnostech (např. pokud uvažujeme třídu **Osoba** a vlastnost věk, její hodnota se každý rok zvyšuje o hodnotu **1**, a to důsledně v ten samý den. Na druhé straně, pokud se podíváme na vlastnosti jako je výška nebo hmotnost tak zjistíme, že tyto vlastnosti se často mění - lidé rostou a mění se jejich hmotnost.

Příklad:

- Pokud pozorujeme pohyb člověka z bodu **A** do bodu **B** a tento pohyb popíšeme v krocích. Například vykročení dopředu, otočení doleva o **45** stupňů, provedení **8** kroků vpřed, otočení doprava o **30** stupňů a provedení dalších **5** kroků vpřed - tyto jednotlivé činnosti můžeme sjednotit do pojmu **metoda**.

Učitel ukáže studentům v prostředí Greenfoot, jak lze zjistit metody, které má třída a které se dají provést na konkrétním objektu.

Učitel ukáže metody definované ve třídě **Actor**. Učitel ukáže, jak se dají zavolat metody na instanci třídy (objektu).

#### 6. Úloha 1.6 (30 min)

##### Cíl:

Studenti se seznámí, jakým způsobem zavolat metodu na instanci třídy (objektu).

##### Pojmy na diskusi:

- volání metod

##### Aktivita:

Učitel popíše úlohu 1.6 Poslat zprávu instanci třídy **Enemy** (zavolat metodu). Učitel otevře poslední verzi projektu. Zatímco studenti postupují podle jeho pokynů a pracují společně krok za krokem.

Učitel odešle zprávu instanci třídy **Enemy**, aby se přesunula na pozici **[12, 6]** a aby byla otočená směrem dolů (kliknutím pravým tlačítkem myši na objekt ve světě, výběrem „**zdeděno z Actor**“ a potom výběrem metody **setLocation(int, int)**). Popíše studentům, co se stane s instancí. Zeptá se, jak byl ovlivněný vnitřní stav příslušné instance.

Učitel demonstruje studentům, jak se definují metody. Vytvoří metodu **setPosition(int x, int y)** na nastavení instance třídy **Actor** na konkrétní souřadnice. Učitel zdůrazní, že tato metoda je ekvivalentní metodě **setLocation(int x, int y)**, a zdůrazní, že je důležité před definováním nové metody zkontrolovat, zda již metoda neexistuje, aby se předešlo duplicitě. Zdůrazňuje, že názvy metod by měly jasně naznačovat jejich účel a funkčnost, což v budoucnu umožní okamžité pochopení funkce metody už jen z jejího názvu. Učitel též zdůrazňuje, že názvy metod by měly být stručné, a uvádí příklady dobře pojmenovaných a špatně pojmenovaných metod. Metody, které může uživatel spustit (zavolat), jsou viditelné po kliknutí pravým tlačítkem myši na příslušnou instanci třídy.

Učitel popíše, jakým způsobem definovat metodu. Studenti postupují podle pokynů učitele a pracují společně krok za krokem.

Učitel požádá studenty, aby definovali metodu, kterou se aktér pohybuje nahoru (zmenšuje se mu souřadnice **y**), a metodu, kterou se pohybuje dolů (zvyšuje se mu souřadnici **y**). Učitel sleduje studenty, jak pracují na úloze, a v případě potřeby dává pokyny.

## 7. Opakování teorie (15 min)

### Cíl:

Studenti rozumí pojmům, které se probíraly.

### Pojmy na diskusi:

- objekt, třída, instance, vnitřní stav, identita, zpráva, metoda.

### Aktivita:

Učitel shrne koncept, který se probíral na vyučování.

## 4. Algoritmus

V rámci tematického celku Algoritmus byly vytvořeny dva metodické materiály

### 4.1. Úvod do algoritmů v prostředí Greenfoot

Tabulka 5. Úvod do algoritmů v prostředí Greenfoot

<b>Název</b>	Úvod do algoritmů v prostředí Greenfoot
<b>Cíle vzdělávání</b>	Na konci této kapitoly by měli studenti důkladně porozumět pojmu algoritmus a algoritmu myšlení, měli by umět navrhovat a implementovat základní algoritmy a měli by být schopni efektivně používat algoritmické koncepty na řešení různých problémů.
<b>Cílová skupina</b>	Studenti středních škol na kurzu OOP4Fun. Studenti by měli mít základní znalosti programování včetně proměnných, funkcí, pojmů iterace a selekce, schopnosti logického uvažování a měli by být schopni řešit problémy. Studenti by měli být seznámeni s programem Greenfoot.
<b>Trvání scénáře</b>	<ol style="list-style-type: none"> <li>1. Úvod do základů algoritmů, algoritmus jako postupnosti kroků (15 min).....</li> <li>2. Úloha 2.1 (20 min) .....</li> <li>3. Algoritmus a jeho vlastnosti (15 min).....</li> <li>4. Úloha 2.2. (25 min).....</li> <li>5. Algoritmizace (15 min).....</li> </ol>

<p><i>Materiály a zdroje</i></p>	<ul style="list-style-type: none"> <li>• Učebnice z projektu OOP4Fun.</li> <li>• Zdroje z projektu OOP4Fun.</li> <li>• Zdrojový kód projektu z repozitáře Github/Gitlab.</li> <li>• Internetové zdroje.</li> </ul>
<p><i>Popis</i></p>	<p>V tomto 90-minutovém výukovém scénáři se studenti středních škol ponoří do oblasti algoritmů a algoritmického myšlení. Výuka začne 15-minutovým úvodem zaměřeným na seznámení studentů se základními pojmy algoritmizace a zdůraznění jejich významu při řešení problémů.</p> <p>Po úvodě se studenti zapojí do 20-minutové úlohy, ve které mají za úkol napsat jednoduchý algoritmus na řešení konkrétního problému. Tato praktická aktivita umožňuje studentům aplikovat pojmy, které byly představeny dříve, a zdokonalit svoje algoritmické dovednosti.</p> <p>Následně bude 15-minutový úsek věnovaný diskusi o algoritmech a jejich vlastnostech. Mezi probíraná témata bude patřit správnost, efektivnost a škálování, přičemž se zdůrazní důležitost jasných a přesných instrukcí při návrhu algoritmů.</p> <p>Na základě svých poznatků stráví studenti dalších 25 minut vytvářením všeobecnějšího algoritmu pro mírně složitý problém. Tato úloha je pro studenty výzvou na abstraktní a kritické myšlení a na uplatňování algoritmických principů při řešení reálných scénářů.</p> <p>V závěrečném 15-minutovém úseku se studenti budou věnovat algoritmizaci, analýze a zdokonalování svých algoritmů. Tento proces zahrne identifikaci potenciálních zlepšení, optimalizaci účinnosti a zabezpečení stability algoritmů.</p> <p>V průběhu celého scénáře budou studenti pracovat individuálně nebo v malých skupinách, což podpoří spolupráci mezi nimi a vzájemné učení. Aktivní účastí na zápisu a analýze algoritmů si studenti rozvinou schopnosti kritického a algoritmického myšlení.</p> <p>Na závěr budou studenti hlouběji rozumět algoritmům a algoritmickému myšlení, což je vybaví základními dovednostmi pro systematický a efektivní přístup ke komplexním problémům.</p>
<p><i>Hodnocení</i></p>	<p>Gamifikace představuje nejen neformální hodnocení, ale také zvýší zájem, vnitřní motivaci a vzdělávací výstupy celé skupiny.</p>
<p><i>Šíření výsledků</i></p>	<p>Na šíření výsledků mezi učiteli a studenty se použije repozitář Github/Gitlab a systém řízení výuky (např. Moodle). Studenti mohou pokračovat v diskusi na dané téma na fóru, které je jim poskytnuté prostřednictvím nástroje na řízení výuky.</p>

#### 4.1.1. Příručka učitele na přípravu vyučovací hodiny

##### 1. Úvod do základů algoritmů, algoritmus jako postupnosti kroků (15 min)

###### Cíl:

Studenti se seznámí s úvodem do algoritmů.

### **Pojmy na diskusi:**

- základy algoritmu

### **Aktivita:**

Učitel představí pojem algoritmus na příkladech z reálného života. Učitel se například zeptá studentů, co dělají ráno od probuzení až po příchod do školy. Potom se učitel studentů zeptá, zda ví, jako se dělá pizza nebo teplý sendvič, anebo zda někdo z nich nezná recept na přípravu jídla nebo koláče.

Učitel spojí proces přípravy jídla nebo koláče s tvorbou programu a zdůrazní, že stejně tak jak existuje recept na přípravu jídla, existuje i „recept“ na tvorbu programu nazývaného algoritmus.

Učitel dospěje k závěru, že algoritmus je soubor kroků, které definují, jakým způsobem se program vykonává.

Učitel dále vysvětlí, že tento soubor kroků se vždy nemusí vykonávat postupně, jeden po druhém, že v algoritmu mohou být i některé kroky, které se mohou vykonávat v závislosti na nějaké podmínce. Učitel požádá studenty, aby uvedli příklad takého případu (například pokud píšeme algoritmus na přípravu teplého sendviče a nemáme nějakou ingredienci, například šunku, ale máme podobnou ingredienci, tak ji můžeme nahradit, nebo dojít do obchodu a chybějící ingredienci sehnat).

Učitel vysvětlí studentům, že některé kroky v algoritmu se mohou několikrát opakovat. Učitel požádá studenty, aby uvedli příklad algoritmu, ve kterém se kroky opakují několikrát.

## 2. Úloha 2.1 (20 min)

### **Cíl:**

Studenti se naučí zapsat jednoduchý algoritmus.

### **Pojmy na diskusi:**

- algoritmus

### **Aktivita:**

Učitel zadá studentům úlohu 2.1, aby na papír napsali postup, kterým popíšu, jakým způsobem chodec přechází přes ulici.

Na začátku učitel studentům neudělí další pokyny, ale sleduje, jak přemýšlejí a pracují. Pokud někdo položí otázku, která je důležitá pro popis postupu přechodu chodce přes ulici, pochválí ho a zdůrazní, z jakého důvodu je tato informace důležitá.

Po určitém čase se učitel zeptá studentů, zda věnovali pozornost tomu, kde chodec přechází přes ulici, zda je to vyznačené místo na přechodě nebo ne. Dále se studentů učitel zeptá, jestli se zamysleli nad tím, zda je na přechodě pro chodce semafor.

Na závěr učitel vybere několik studentů, kteří by měli přečíst své pokyny na přechod přes silnici.

## 3. Algoritmus a jeho vlastnosti (15 min)

### **Cíl:**

Studenti se seznámí s vlastnostmi algoritmu

### Pojmy na diskusi:

- vlastnosti algoritmu

### Aktivita:

Učitel dále studentům vysvětlí vlastnosti algoritmu. Učitel studentům také ukáže, že algoritmy je možné znázornit i graficky, a uvede příklad graficky znázorněných algoritmů.

#### 4. Úloha 2.2. (25 min)

##### Cíl:

Student dokáže zapsat všeobecnější algoritmus

### Pojmy na diskusi:

- zápis algoritmu

### Aktivita:

Učitel vyzve studenty, aby vytvořili všeobecný algoritmus na přípravu horkého nápoje. Dále studenty upozorní, aby si rozmysleli, jaké musí být vstupy takého algoritmu, aby byl všeobecný.

#### 5. Algoritmizace (15 min)

##### Cíl:

Studenti dovedou naprogramovat více příkladů a umějí se zapojit do definování vlastních

### Pojmy na diskusi:

- algoritmus

### Aktivita:

Učitel studentům vysvětlí, že i v matematice existují určité algoritmy, které používáme při řešení problémů. Učitel se studentů zeptá, zda by mohli uvést nějaký příklad.

Příklad, který učitel studentům vysvětlí, je příklad výpočtu hodnoty aritmetického výrazu, který má několik matematických operací, kde je potřebné dodržovat pravidla a priority vykonávání matematických operací.

Učitel uvede i další příklady, například montáž nového nábytku, který byl zakoupen a přišel s návodem, který popisuje, jak tento nábytek smontovat. Příkladem mohou být také pokyny, které dostáváme prostřednictvím GPS navigace, pokud se chceme dostat z bodu **A** do bodu **B**.

Učitel studenty požádá, aby na papír napsali vlastní algoritmus, poté jej někteří z nich odprezentují

## 4.2. Dobrodružství v prostředí Greenfoot: Volání metod v Javě, práce s dokumentací a řízení aplikace

Tabulka 6. Dobrodružství v prostředí Greenfoot: Volání metod v Javě, práce s dokumentací a řízení aplikace

<b>Název</b>	Dobrodružství v prostředí Greenfoot: Volání metod v Javě, práce s dokumentací a řízení aplikace
<b>Cíle vzdělávání</b>	Na konci této kapitoly by studenti měli dobře porozumět volání metod v jazyce Java, zejména se zaměřením na metody <b>act()</b> a <b>move()</b> v prostředí Greenfoot. Měli by ovládat efektivní

	<p>používání klíčového slova <b>this</b> jako referenci na aktuální objekty v kontextu třídy. Kromě toho by studenti měli prokázat schopnost zavolat metody v rámci třídy, pochopit syntaxi a parametry potřebné na volání metod. Také by měli prokázat dovednosti s používáním technik volání metod na efektivními řešení úloh v rámci interaktivního vývoje her. Navíc by měli pochopit důležitost dokumentace kódu programu a měli by být schopni efektivně dokumentovat program v jazyce Java a zabezpečit jeho přehlednost a čitelnost. Nakonec by měli ovládat techniky řízení aplikací v prostředí Greenfoot, které jim umožní přesnou manipulaci a interakci s herními prvky na vytvoření poutavé a funkční herní mechaniky.</p>
<i>Cílová skupina</i>	<p>Studenti středních škol na kurze OOP4Fun. Základní znalosti programování včetně konceptů iterace a větvení. Studenti by měli být seznámeni s programem Greenfoot.</p>
<i>Trvání scénáře</i>	<ol style="list-style-type: none"> <li>1. Vysvětlení metody <code>act</code> (10 min) .....</li> <li>2. Vysvětlení metody <code>move</code> (20 min) .....</li> <li>3. Zavedení klíčového slova <code>this</code> (5 min) .....</li> <li>4. Úloha 2.3. (10 min) .....</li> <li>5. Vysvětlení automatického doplnění (5 min) .....</li> <li>6. Význam dokumentace zdrojového kódu programu (15 min) .....</li> <li>7. Úloha 2.4.(5 min) .....</li> <li>8. Úloha 2.5. (5 min) .....</li> <li>9. Úloha 2.6. (10 min) .....</li> <li>10. Úloha 2.7. (20 min) .....</li> <li>11. Diskuse: algoritmus a jeho vlastnosti, algoritmizace, tlačítka v prostředí Greenfoot (5 min) .....</li> </ol>
<i>Materiály a zdroje</i>	<ul style="list-style-type: none"> <li>● učebnice z projektu OOP4Fun.</li> <li>● zdroje z projektu OOP4Fun.</li> <li>● zdrojový kód projektu z repozitáře Github/Gitlab.</li> <li>● internetové zdroje.</li> </ul>
<i>Popis</i>	<p>V tomto 110-minutovém výukové scénáři se studenti středních škol vydají na cestu ke zvládnutí volání metod jazyka Java, dokumentace programu a řízení aplikace v prostředí Greenfoot.</p> <p>Hodina začne 10-minutovým zkoumáním metody <code>act()</code>, po kterém následuje 20 minutový úvod do metody <code>move()</code>, která je základní součástí programování v prostředí Greenfoot.</p> <p>Potom se studenti budou 5 minut věnovat významu klíčového slova <b>this</b> při odkazování na aktuální objekty v kontextu třídy.</p> <p>Potom na studenty čeká 10-minutová úloha, která je vyzve, aby si procvičili volání metod v rámci třídy, přičemž použijí syntaxi a parametry potřebné na volání metod.</p> <p>Následuje 5-minutové vysvětlení funkcí automatického dokončování v prostředí Greenfoot, které se zaměřuje na zvýšení efektivnosti programování.</p> <p>V době dalších 15 minut studenti pochopí důležitost dokumentace programu, zejména jak jasná a stručná dokumentace zvyšuje čitelnost a udržitelnost programu.</p> <p>V 5-minutové úloze studenti doplní dokumentaci ke svému kódu, čímž zabezpečí jasnost a srozumitelnost pro sebe a ostatních.</p> <p>V návaznosti na toto úlohu stráví studenti dalších 5 minut přidáváním podrobnější</p>



	<p>dokumentace ke svému kódu.</p> <p>V další 10-minutové úloze studenti budou zkoumat a číst dokumentaci přidanou jejich vrstevníky, čím získají přehled o různých stylech a přístupech k programování.</p> <p>Nakonec studenti před závěrečnou 5-minutovou diskusí stráví 20 minut zkoumáním technik řízení aplikací v prostředí Greenfoot, které jim umožní přesnou manipulaci a interakci s herními prvky na vytvoření poutavé a funkční herní mechaniky.</p> <p>V průběhu celé doby budou studenti pracovat individuálně nebo v malých skupinách, což podpoří spolupráci a vzájemné učení. Aktivní účastí na zápise a analýze kódu si studenti rozvinou schopnosti kritického myšlení a řešení algoritmických problémů.</p> <p>Na závěr studenti získají hlubší pochopení z oblasti volání metod jazyka Java, dokumentace kódu a řízení průběhu aplikací v prostředí Greenfoot, čímž budou vybaveni základními dovednostmi pro vývoj her i mimo něj.</p>
<i>Hodnocení</i>	Gamifikace představuje nejen neformální hodnocení, ale také zvýší zájem, vnitřní motivaci a vzdělávací výstupy celé skupiny.
<i>Šíření výsledků</i>	Na šíření výsledků mezi učiteli a studenty se použije repozitář Github/Gitlab a systém řízení výuky (např. Moodle). Studenti mohou pokračovat v diskusi na dané téma na fóru, které je jim poskytnuté prostřednictvím nástroje na řízení výuky.

#### 4.2.1. Příručka učitele na přípravu vyučovací hodiny

##### 1. Vysvětlení metody `act` (10 min)

###### Cíl:

Studenti porozumějí použití metody `act` a volání metody z jiné metody.

###### Pojmy na diskusi:

- metoda `act`

###### Aktivita:

Učitel otevře poslední verzi projektu **TowerDefense**. Učitel umístí instance třídy **Enemy** na pozici **[0,0]**. Učitel se studentů zeptá, co se stane, pokud zavoláme metodu `act()` na instanci třídy **Enemy**. Je to očekávané chování?

Učitel požádá studenta, aby mu pomohl splnit úlohu, posunout objekt třídy **Enemy** o dvě buňky v aktuálním směru při zavolání metody `act()`.

Učitel studentům ukáže, jakým způsobem toto úlohu provést (přidá volání metody `move(int)` do těla metody `act()`), přičemž dává pokyny a studenti sdělují své nápady.

##### 2. Vysvětlení metody `move` (20 min)

###### Cíl:

Vysvětlit metodu `move` a způsoby pohybu instancí vpřed a vzad.

###### Pojmy na diskusi:

- metody pro pohyb

**Aktivita:**

Učitel vysvětlí metodu **move(int)**. Učitel umístí instance třídy **Enemy** na různé pozice a zavolá metodu s tím, že za parametry zadá kladné hodnoty, například **1** nebo **3**. Učitel se studentů zeptá, co si myslí, že se stane po zavolání metodu **move()** a zadání záporné hodnoty, například **-1**.

Učitel studentům zadá úlohu napsat metodu **backward()**, která posune instanci třídy o **1** krok dozadu. Učitel se zeptá, zda je možné posunout objekt o **2** kroky dozadu a poté o **x** kroků vpřed a co vše je k tomu nutné provést.

3. Zavedení klíčového slova **this** (5 min)

**Cíl:**

Studenti porozumí zavedení klíčového slova **this**

**Pojmy na diskusi:**

- klíčové slovo **this**

**Aktivita:**

Učitel zavede klíčové slovo **this**.

4. Úloha 2.3. (10 min)

**Cíl:**

Studenti budou vědět, jakým způsobem napsat metodu na vertikální pohyb objektu, která mění hodnotu osy **y**.

**Pojmy na diskusi:**

- volání metody

**Aktivita:**

- Učitel se zeptá studentů, jakým způsobem lze s instancí třídy pohybovat vertikálně, nahoru nebo dolů. Učitel požádá studenta, aby našel vhodnou metodu pro pohyb nahoru/dolů tím, že klikne pravým tlačítkem myši na instanci třídy vybere „**zděděno z Actor**“ a poté vybere vhodnou metodu. Student by si měl všimnout metod: **turn**, **setRotation**, **setLocation**, **getLocation**, **getRotation**. Učitel poté zadá studentům úlohu napsat kód metod **up()** a **down()**. Učitel sleduje práci studentů a na závěr spolu s nimi vysvětlí, jakým způsobem tyto metody implementovat. Metody **up()** a **down()** mění hodnotu souřadnice **y** tím, že ji zvyšují nebo snižují. V tomto okamžiku může učitel představit parametry metod, ale zatím bez podrobností.

5. Vysvětlení automatického doplnění (5 min)

**Cíl:**

Studenti se seznámí s automatickým doplňováním v prostředí Greenfoot

**Pojmy na diskusi:**

- automatické doplňování (autocomplete), **CRTL+SPACE**

**Aktivita:**

Učitel popíše studentům, jakým způsobem najít metodu na instanci třídy, kterou lze zavolat v případě, že např. zapomněli název nebo např. pokud právě začínají programovat a chtějí požádat prostředí Greenfoot, aby jim pomohlo dokončit psaní kódu.

#### 6. Význam dokumentace zdrojového kódu programu (15 min)

##### Cíl:

Studenti se seznámí s komentáři dokumentací

##### Pojmy na diskusi:

- komentáře a dokumentace, prozkoumání okna dokumentace.

##### Aktivita:

Učitel studenty seznámí s příkazem v programu, který není součástí vykonávání programu (komentáře, dokumentační komentáře).

Učitel musí zdůraznit rozdíly mezi komentářem a dokumentací (jako speciálním typem komentáře).

Učitel studentům představí zdrojový kód třídy **Enemy** a následně popíše komentář, který popisuje dokumentaci třídy **Enemy**. Učitel tu popíše, že existuje určité pravidlo, které musíme dodržovat při psaní dokumentace pro naši třídu nebo metody.

Učitel studentům zadá úlohu, aby prozkoumali, jakým způsobem psát dokumentaci ke třídám a metodám v jazyce Java. Vysvětlí studentům, že v menu prostředí Greenfoot je volba „**upravit -> doplnit dokumentační komentář metody**“

#### 7. Úloha 2.4.(5 min)

##### Cíl:

Studenti se seznámí s dokumentačními komentáři k metodě

##### Pojmy na diskusi:

- dokumentační komentář k metodě

##### Aktivita:

Učitel studentům zadá úlohu přidat komentář k dokumentaci metody **act()**.

**Commit:** [68b1c82c7df2c7826f2d3f78373498569adab7e9](https://github.com/68b1c82c7df2c7826f2d3f78373498569adab7e9)

#### 8. Úloha 2.5. (5 min)

##### Cíl:

Studenti se seznámí s dokumentačními komentáři ke třídě

##### Pojmy na diskusi:

- dokumentační komentář ke třídě

##### Aktivita:

Učitel studentům zadá úlohu přidat dokumentační komentář k metodě **act()**.

Učitel vyzve studenty k úpravě komentáře k dokumentaci třídy **Enemy**. Konkrétně k přidání verze třídy a jejího autora.

Commit: [1a7a9f83c5271a7c0dfa46ce3b1ee65682b0c5e5](#)

#### 9. Úloha 2.6. (10 min)

**Cíl:**

Studenti se naučí prozkoumat dokumentaci třídy s cílem pochopit chování metody

**Pojmy na diskusi:**

- prozkoumání dokumentace

**Aktivita:**

Učitel studentům zadá úlohu prozkoumat okno s dokumentací a přečíst si dokumentaci ke třídám **Actor** a **World** (kliknutím pravým tlačítkem na příslušné třídy a volbou „**otevřít dokumentaci**“). Učitel zdůrazní důležitost čtení dokumentace s cílem najít metody, které mohou být užitečné.

#### 10. Úloha 2.7. (20 min)

**Cíl:**

Studenti se seznámí s tlačítky v prostředí Greenfoot

**Pojmy na diskusi:**

- Tlačítka v prostředí Greenfoot

**Aktivita:**

Učitel pokračuje v práci na poslední verzi projektu. Učitel požádá studenty, aby přidali dvě instance objektu třídy **Enemy** a zavolali metodu **act()** na každé z nich.

Učitel popíše tlačítko **Krok (Act)** umístěné v hlavním okně. Učitel na něj klikne. Poté studentům položí otázku, aby vysvětlili, co se stalo.

Učitel také požádá studenta, aby kliknul na tlačítko **Spustit (Run)** a popsal, co se stalo.

Učitel požádá studenty, aby klikli na tlačítko **Reset** a vysvětlili, co se stalo. Potom učitel vysvětlí, co je nutné udělat, aby se při každém kliknutí na tlačítko **Reset** ve světě objevily dvě instance třídy **Enemy** na pozicích **[0,3]** a **[3,3]**.

Učitel by studentům měl vysvětlit, že pokud chtějí, aby se tyto instance objevily ve světě vždy po kliknutí na tlačítko **Reset**, je nutné změnit konstruktor třídy **World** tak, aby se tyto instance vytvořili v konstruktoru a umístily na požadované pozice.

#### 11. Diskuse: algoritmus a jeho vlastnosti, algoritmizace, tlačítka v prostředí Greenfoot (5 min)

**Cíl:**

Studenti dokážou shrnout to, co se naučili v této kapitole.

**Pojmy na diskusi:**

metody pohybu, tlačítka prostředí Greenfoot, dokumentace

**Aktivita:**

Učitel shrne učivo. Učitel tu může zdůraznit důležitost zdrojového kódu, pravidla pro pojmenování metod a tříd.

## 5. Větvení

V rámci tematického celku Větvení byly vytvořené dva výukové scénáře.

### 5.1. Zkoumání větvení při vývoji her v prostředí Greenfoot - Neúplné větvení kódu

Tabulka 7. Zkoumání větvení při vývoji her v prostředí Greenfoot - Neúplné větvení kódu

<i>Název</i>	Zkoumání větvení při vývoji her v prostředí Greenfoot - Neúplné větvení kódu
<i>Cíle vzdělávání</i>	Kapitola zahrnuje neúplné větvení (vícenásobné větvení je záměrně vynechané). Studentům představí základy vnímání světa hercem. Studenti budou schopni psát kód s využitím podmínek. Po ukončení této kapitoly se studenti naučí používat jednoduché příkazy <b>if-else</b> a naučí se v kódu vyvářet rozhodování, které bude kontrolovat chování jejich hry. Získají základní znalosti programovacího jazyka Java, naučí se potřebnou syntaxi, analyzovat a pochopit kód, což jim pomůže přijít na to, z jakého důvodu se jejich hra chová určitým způsobem a jak odstranit problémy. Studenti budou schopni vytvořit vlastní projekty her s využitím toho, co se naučili o objektivě orientovaném programování.
<i>Cílová skupina</i>	Studenti středních škol na kurzu OOP4Fun. Základní znalosti programování včetně proměnných a datových typů. Studenti by měli být seznámeni s programem Greenfoot.
<i>Trvání scénáře</i>	<ol style="list-style-type: none"> <li>1. Úvod (5 min) .....</li> <li>2. Vysvětlení kódu (15 min) .....</li> <li>3. Neúplné větvení (10 min) .....</li> <li>4. Pozorování stavu hráče (10 min) .....</li> <li>5. Úloha 3.2 - Přidání detekce okraje světa (10 min) .....</li> </ol>
<i>Materiály a zdroje</i>	<ul style="list-style-type: none"> <li>● Učebnice z projektu OOP4Fun.</li> <li>● Zdroje z projektu OOP4Fun.</li> <li>● Zdrojový kód projektu z repozitáře Github/Gitlab.</li> <li>● Internetové zdroje.</li> </ul>
<i>Popis</i>	<p>Výukový scénář začne 5-minutovým úvodem, kde učitel představím větvení na reálné úloze ze života. Poté hodina pokračuje 15-minutovým vysvětlením kódu metody <b>turn()</b>, které připravuje půdu pro pochopení neúplného větvení kódu. Tuto diskusi vede učitel a jejím cílem je přivést studenty ke společnému pochopení úlohy této metody ve světě herců.</p> <p>Pak následuje 10-minutový blok věnování osvojení si základních pojmů souvisejících s neúplným větvením. Tato fáze osvojení je pro studenty klíčová, aby si osvojili základní principy předtím, než se pustí do psaní kódu.</p> <p>Následně se studenti věnují 10-minutové úloze zkoumání, ve které pozorují stav herce v rámci svého kódu, což podporuje hlubší pochopení běhu programu na základě zdrojového kódu.</p> <p>Následující 10-minutová produkční fáze zahrnuje přidání detekce okrajů světa do jejich</p>

	projektů, přidání chování a nastavení hry.
<i>Hodnocení</i>	<p>Tato aktivita umožní učitelům poskytovat formativní zpětnou vazbu na základě diskusí a monitorování žáků v obrácené třídě.</p> <p>Vzájemné hodnocení se bude vykonávat online jako součást domácí úlohy. Studentům to připomene důležité aspekty úlohy, přinutí je kriticky hodnotit práci jiných studentů, poskytne jim to přehled o dobrých nebo ne velmi dobrých řešeních jejich vrstevníků a zvýší to celkové dosažení výsledků vzdělávání.</p> <p>Při práci na týmovém projektu se budou využívat také tyto výsledky vzdělávání a poznatky.</p>
<i>Šíření výsledků</i>	Na šíření výsledků mezi učiteli a studenty se použije repozitář Github/Gitlab a systém řízení výuky (např. Moodle). Studenti mohou pokračovat v diskusi na dané téma na fóru, které je jim poskytnuto prostřednictvím nástroje na řízení výuky.

### 5.1.1. Příručka učitele na přípravu vyučovací hodiny

#### 1. Úvod (5 min)

##### Cíl:

Učitel představí cíle této vyučovací hodiny.

##### Pojmy na diskusi:

- algoritmus, dokumentace kódu.

##### Aktivity:

Učitel představí novou látku, která se bude na dnešní hodině probírat. Na ilustraci dnešní hodiny učitel uvede příklad algoritmu s jednoduchým větvením. Příkladem může být algoritmus přechodu přes ulici na přechodu pro chodce, kde není semafor. Chodec nepřejde ulici hned, ale nejdříve se ujistí, že zleva ani zprava nepřijíždějí žádná vozidla. Pokud se nepřibližují žádná vozidla, chodec přejde přes ulici.

#### 2. Vysvětlení kódu (15 min)

##### Cíl:

##### Pojmy na diskusi:

##### Aktivity:

Učitel stáhne nejnovější verzi projektu:

- z platformy Moodle
- z GIT repozitáře.

Učitel vytvoří instanci třídy **Enemy** a její instanci a umístí někde do světa. Dále vysvětlí některé metody společné třídy **Actor**:

- **move(int)**
- **turn(int)**
- **setRotation()**

Při vysvětlování metod učitel ukazuje, jakým způsobem se mění některé vlastnosti instance třídy (například poloha objektu ve světě, tj. hodnoty atributů **x** a **y**). Učitel diskutuje se studenty o tom, jakým způsobem doplnit metodu **act()** tak, aby se při každém jejím volání posunula instance třídy **Enemy** o dva kroky dopředu.

### 3. Neúplné větvení (10 min)

**Cíl:**

**Pojmy na diskusi:**

- větvení, neúplné větvení.

**Aktivita:**

Učitel pokračuje v práci na projektu. Učitel umístí do světa instanci třídy **Enemy**. Učitel vysvětlí studentům, jakým způsobem mohou zkontrolovat, zda se objekt nachází v horní polovině světa, a zobrazí správu „Nalezen“.

Učitel vysvětlí studentům metodu **showText()**, která se používá na zobrazení textu.

### 4. Pozorování stavu hráče (10 min)

**Cíl:**

**Pojmy na diskusi:**

- vnitřní stav instance.

**Aktivita:**

Učitel vytvoří instanci třídy **Enemy** a umístí ji do středu světa. Učitel otevře okno s vnitřním stavem instance a umístí ho tak, aby bylo viditelné v průběhu spuštění aplikace. Potom spustí aplikaci a třída pozoruje, jak se mění hodnoty atributů **x**, **y** a **rotation** v instanci třídy **Enemy** při volání různých metod. Dále s třídou diskutuje, jakým způsobem by se tyto hodnoty měnily při pohybu (nahoru, dolů, vlevo a vpravo) a při otáčení.

### 5. Úloha 3.2 - Přidání detekce okraje světa (10 min)

**Cíl:**

**Pojmy na diskusi:**

- detekce okraje světa.

**Aktivita:**

Učitel diskutuje se studenty o tom, jakým způsobem mohou určit, zda je předmět na okraji světa nebo ne. Například, pokud známe rozměry stolu, na základě polohy (**x** a **y**) se dá určit, zda je předmět na okraji stolu nebo není.

Učitel umístí instanci třídy **Enemy** někde ve světě (ale ne na jeho okraj) a zavolá metodu **isAtEdge()**. Diskutuje se studenty o tom, co se stalo. Učitel přesune instanci na okraj a znovu prozkoumá výsledek metody **isAtEdge()**, která by teď měla vrátit hodnotu **true**.

Učitel vysvětlí metodu **isAtEdge()**.

Učitel zadá studentům úlohu 3.2 - přidat do těla metody **act()** kód, který otočí instanci třídy **Enemy** o 180° voláním metody **setRotation()**, v okamžiku kdy dosáhne okraj světa.

Učitel se studenty prodiskutuje, jakým způsobem může objekt, který dosáhnul okraj světa, pokračovat ve svém pohybu:

- dozadu (bez otočení)
- dozadu (s otočením)

Učitel spolu se studenty doplní programový kód metody **act()**.

Po spuštění metody **act()** se studenty diskutuje o tom, co se stane s instancí třídy **Enemy** v okamžiku, kdy dosáhne okraj světa.

**Commit:** [4927c3ff7eb39b51ba2738f2ab500fd6c32e3bb4](https://github.com/4FUN/4FUN/commit/4927c3ff7eb39b51ba2738f2ab500fd6c32e3bb4)

## 5.2. Zkoumání větvení při vývoji her v prostředí Greenfoot - Úplné větvení kódu

Tabulka 8. Zkoumání větvení při vývoji her v prostředí Greenfoot - Úplné větvení kódu

<b>Název</b>	Zkoumání větvení při vývoji her v prostředí Greenfoot - Úplné větvení kódu
<b>Cíle vzdělávání</b>	Kapitola zahrnuje neúplné a úplné větvení. Zavádí nové způsoby vnímání světa hercem. Studenti budou schopni psát kód s použitím složitějších podmínek. Po ukončení této kapitoly se studenti naučí používat nové možnosti příkazů <b>if-else</b> a naučí se v kódu vytvářet nové specifitější a sofistikovanější způsoby rozhodování, které bude kontrolovat chování jejich hry. Získají další základní znalosti programovacího jazyka Java, naučí se potřebnou syntaxi, analyzovat a pochopit složitější kód, což jim pomůže přijít na to, z jakého důvodu se jejich hra chová určitým způsobem a jak odstranit problémy. Studenti budou schopni vytvořit vlastní projekty her s využitím nových poznatků, které se naučili.
<b>Cílová skupina</b>	Studenti středních škol na kurzu OOP4Fun. Základní znalosti programování včetně proměnných a datových typů. Studenti by měli být seznámeni s programem Greenfoot.
<b>Trvání scénáře</b>	<ol style="list-style-type: none"> <li>1. Úloha 3.3 - Přidání tříd <b>Direction</b> a <b>Orb</b> (30 min) .....</li> <li>2. Vysvětlení detekce kolizí (30 min) .....</li> <li>3. Úloha 3.4 - Přidání detekce kolizí (10 min) .....</li> <li>4. Úloha 3.5 - Předvídání pohybu nepřítele ve vlastním uspořádání (15 min) .....</li> <li>5. Úloha 3.6 - Předvídání pohybu nepřítele při specifickém uspořádání (15 min) .....</li> <li>6. Vysvětlení kódu: Úplné větvení (15 min) .....</li> <li>7. Úloha 3.7 – Použití úplného větvení při detekci kolizí (20 min) .....</li> <li>8. Úloha 3.8 - Předvídání pohybu nepřítele na základě předchozích nastavení (30 min) .....</li> <li>9. Opakování teorie (5 min) .....</li> </ol>
<b>Materiály a zdroje</b>	<ul style="list-style-type: none"> <li>• Učebnice z projektu OOP4Fun.</li> <li>• Zdroje z projektu OOP4Fun.</li> <li>• Zdrojový kód projektu z Github/Gitlab.</li> <li>• Internetové zdroje.</li> </ul>
<b>Popis</b>	170-minutový výukový scénář začne 30-minutovým cvičením na přidání tříd <b>Direction</b> a <b>Orb</b> . To pomůže studentům pochopit objektivě orientovanou povahu jazyka Java a důležitost správné struktury kódu. Následuje 30-minutové vysvětlení kódu zaměřené na koncept detekce kolizí, který pomáhá studentům pochopit interakci objektů v rámci jejich herního



	<p>prostředí. Potom studenti stráví 10 minut přidáváním detekce kolizí do svého projektu v další produkční úloze, přičemž využijí své znalosti větvení.</p> <p>Následují dvě výzkumné úlohy v trvání 15 minut, ve kterých mají studenti za úlohu předvídat pohyb nepřítele ve vlastním a specifickém uspořádání světa, což posiluje jejich schopnosti řešit problémy a analytické dovednosti. Další 15-minutová diskuse je zaměřená na úplné větvení, čímž se zabezpečuje, aby studenti uměli rozlišovat mezi neúplným a úplným větvením kódu.</p> <p>20-minutová produkční úloha zapojí studenty do používání úplného větvení s detekcí kolizí, přičemž cílem je upevnit jejich učivo aplikováním komplexních konceptů v praktickém prostředí. Hodinu uzavírá náročná 30-minutová investigativní úloha, ve které studenti opět předpovídají pohyb nepřítele, tentokrát ale s novými zkušenostmi z předchozích úloh, čímž uplatňují své kumulované vědomosti.</p> <p>Vyučování končí 5-minutovým teoretickým zopakováním naučených pojmů.</p>
<i>Hodnocení</i>	<p>Tato aktivita umožní učitelům poskytovat formativní zpětnou vazbu na základě diskusí a monitorování obrácené třídy a týmové práce studentů.</p> <p>Vzájemné hodnocení bude probíhat online jako součást domácí úlohy. Studentům se tak připomenou důležité aspekty cvičení, donutí se kriticky posoudit práci ostatních studentů, získají přehled o dobrých či méně dobrých řešeních svých spolužáků a zvýší se celkové dosáhnutí výsledků vzdělávání.</p> <p>Při práci v týmovém projektu, na kterém studenti pracují, se budou využívat také tyto výsledky vzdělávání a poznatky.</p>
<i>Šíření výsledků</i>	<p>Na šíření výsledků mezi učiteli a studenty se použije repozitář Github/Gitlab a systém řízení výuky (např. Moodle). Studenti mohou pokračovat v diskusi na dané téma na fóru, které je jim poskytnuté prostřednictvím nástroje na řízení výuky.</p>

### 5.2.1. Příručka učitele na přípravu vyučovací hodiny

#### 1. Úloha 3.3 - Přidání tříd Direction a Orb (30 min)

##### Cíl:

Pochopit, jakým způsobem přidat novou třídu do projektu.

##### Pojmy na diskusi:

- třída.

##### Aktivity:

Učitel zadá studentům úlohu 3.3. Učitel sleduje činnost studentů a na závěr požádá jednoho studenta, aby prezentoval svojí práci. Student popíše a představí svoji práci.

**Commit:** [4ed6b37e6d481181d8b340639aa03391406b6c2e](https://github.com/4FUN/4FUN/commit/4ed6b37e6d481181d8b340639aa03391406b6c2e)

#### 2. Vysvětlení detekce kolizí (30 min)

##### Cíl:

Vysvětlení detekce kolizí.

**Pojmy na diskusi:**

- detekce kolizí.

**Aktivita:**

Učitel umístil instanci třídy **Enemy** do světa a instanci třídy **Direction** do toho stejného řádku. Učitel přidá kód do metody **act()** tak, aby se objekt posunul o jeden krok dopředu.

Učitel vysvětlí studentům, jak určit, zda se dva nebo více objektů („herců“) ve světě nachází na stejné pozici (ve stejné buňce). Učitel vysvětlí metodu **isTouching()**.

Učitel a studenti upraví metodu **act()** třídy **Enemy** tak, aby zabezpečila, že se nepřítel otočí o **90°** ve směru hodinových ručiček, pokud se nachází ve stejné buňce, která obsahuje instanci třídy **Direction**.

Učitel spolu se studenty pozoruje, co se děje s atributem **rotation**.

3. [Úloha 3.4 - Přidání detekce kolizí \(10 min\)](#)

**Cíl:**

Studenti rozumějí, jakým způsobem určit, zda jsou dva objekty na stejné pozici (ve stejné buňce).

**Pojmy na diskusi:**

**Aktivita:**

Učitel zadá studentům úlohu 3.4. Učitel sleduje činnost studentů a na závěr požádá jednoho z nich, aby prezentoval svoji práci. Student popíše a představí svoji práci.

**Commit:** [968e6f195e3def25e11bc41b664ba1715f7da11d](#)

4. [Úloha 3.5 - Předvídání pohybu nepřítele ve vlastním uspořádání \(15 min\)](#)

**Cíl:**

Studenti rozumějí pohybu objektů.

**Pojmy na diskusi:**

- pohyb objektů.

**Aktivita:**

Učitel zadá studentům úlohu 3.5. Učitel sleduje činnost studentů a na závěr požádá jednoho z nich, aby prezentoval svoji práci. Student popíše a představí svoji práci.

5. [Úloha 3.6 - Předvídání pohybu nepřítele při specifickém uspořádání \(15 min\)](#)

**Cíl:**

Studenti rozumějí pohybu objektů.

**Pojmy na diskusi:**

- pohyb objektů.

**Aktivita:**

Učitel zadá studentům úlohu 3.6. Učitel sleduje činnost studentů a na závěr požádá jednoho z nich, aby prezentoval svoji práci. Student popíše a představí svoji práci.

#### 6. Vysvětlení kódu: Úplné větvení (15 min)

##### Cíl:

Učitel vysvětlí, kdy a jak používat příkazy **if-else** a **switch**.

##### Pojmy na diskusi:

- příkaz **if-else**, vnořené příkazy **if**, příkaz **switch**.

##### Aktivity:

Učitel zadá studentům úlohu, aby popsali na papír, jakým způsobem chodec přechází přes ulici. Učitel sleduje, jak studenti řeší úlohu. V určitém momentě učitel zdůrazní studentům, aby věnovali pozornost tomu, zda na místě přechodu přes ulici je či není je semafor. Pokud mezi tím, tedy před zdůrazněním, jeden ze studentů položí tuto nebo podobnou otázku, tak ho učitel pochválí za aktivitu a zdůrazňuje, že je důležité, abychom před programováním vždy nejdříve analyzovali problém a identifikovali všechny případy, ke kterým může dojít. Zde by studenti měli pochopit úplné a neúplné větvení.

Učitel požádá studenty, aby umístili instance třídy **Orb** a **Direction** doleva, doprava k různým hranicím světa. Studenti popíšu „špatné“ chování instance třídy **Enemy**. Učitel by měl vysvětlit, že v jeden okamžik jsou splněny dvě podmínky, kdy se instance třídy **Enemy** dotýká instance tříd **Orb/Direction** a zároveň se dotýká okraje světa. Nakreslením situace na tabuli nebo papír studenti zjistí, že je potřebné úplné a vnořené větvení a potom studenti změní kód metody **act()** instance třídy **Enemy**.

#### 7. Úloha 3.7 – Použití úplného větvení při detekci kolizí (20 min)

##### Cíl:

Studenti rozumí úplnému větvení, vnořeným příkazům **if** a příkazu **switch**.

##### Pojmy na diskusi:

- vnořené příkazy **if**, příkaz **switch**.

##### Aktivity:

Učitel zadá studentům úlohu 3.7. Učitel sleduje činnost studentů a na závěr požádá jednoho z nich, aby prezentoval svoji práci. Student popíše a představí svoji práci.

**Commit:** [f017de8b49d4fc77f62afac4d842429560bcfb8b](https://github.com/f017de8b49d4fc77f62afac4d842429560bcfb8b)

#### 8. Úloha 3.8 - Předvídání pohybu nepřítele na základě předchozích nastavení (30 min)

##### Cíl:

Studenti rozumí jakým způsobem ovlivňuje větvení v metodě **act()** chování instancí třídy **Enemy** a dokáží tak předvídat pohyb nepřítele na základě předchozích nastavení.

### Pojmy na diskusi:

- chování objektů.

### Aktivita:

Učitel libovolně vkládá instance do světa a studenti vysvětlují jejich pohyb a chování (samostatně nebo ve dvojicích). Učitel zadá studentům úlohu 3.8.

## 9. Opakování teorie (5 min)

### Cíl:

Opakování teorie v souvislosti s termíny, které byly probrány

### Pojmy na diskusi:

- principy větvení.

### Aktivita:

V této části zopakuje koncepty diskutované v rámci této celé kapitoly.

## 6. Proměnné a výrazy

V rámci tematického celku Proměnné a výrazy bylo vytvořeno pět vyučovacích scénářů.

### 6.1. Úvod do proměnných a datových typů v prostředí Greenfoot

Tabulka 9. Úvod do proměnných a datových typů v prostředí Greenfoot

<b>Název</b>	Úvod do proměnných a datových typů v prostředí Greenfoot
<b>Cíle vzdělávání</b>	Na konci této kapitoly budou studenti schopni porozumět datovým typům a proměnným. Pojmy budou probrány v kontextu vývoje hry, čímž se podpoří kreativita, týmová práce a nadšený přístup k programování v prostředí Greenfoot.
<b>Cílová skupina</b>	Studenti středních škol na kurzu OOP4Fun. Základní znalosti programování včetně proměnných a datových typů. Studenti by měli být seznámeni s programem Greenfoot.
<b>Trvání scénáře</b>	<ol style="list-style-type: none"> <li>1. Úvod (10 min) .....</li> <li>2. Identifikace proměnné (5 min) .....</li> <li>3. Typy údajů (15 min) .....</li> <li>4. Deklarace proměnných (10 min) .....</li> <li>5. Inicializace proměnných (5 min) .....</li> </ol>
<b>Materiály a zdroje</b>	<ul style="list-style-type: none"> <li>• Učebnice z projektu OOP4Fun.</li> <li>• Zdroje z projektu OOP4Fun.</li> <li>• Zdrojový kód projektu z Github/Gitlab.</li> <li>• Internetové zdroje.</li> </ul>
<b>Popis</b>	V tomto 45-minutovém výukovém scénáři se studenti středních škol seznámí se zásadami programování, které se týkají datových typů a proměnných z pohledu vývoje her ve vývojovém prostředí Greenfoot. Vyučování začne 10-minutovým úvodem učitele, ve kterém

	<p>se vytvoří kontext související s předchozími vyučovacími hodinami, což vytvoří základ pro zavedení a definování proměnných.</p> <p>Potom následuje 5-minutový scénář, během kterého studenti a učitelé zkoumají a určují proměnné pro svoji hru. Vzhledem k faktu, že každá proměnná musí být určitého typu, se v následujících 15 minutách odprezentují různé typy proměnných.</p> <p>Základní aktivity zahrnují 10-minutové cvičení pod vedením učitele, během kterého studenti deklarují proměnné pro svoji hru, přičemž je zdůrazněna důležitost názvů a typů proměnných. Potom následuje 5-minutové cvičení týkající se inicializace proměnných, během kterého se definují hodnoty proměnných. V této souvislosti se může měnit chování objektů ve hře (např. otáčení objektu, pohyb objektu).</p> <p>Studenti budou pokračovat v práci na hře, která byla vysvětlená a započatá dříve. Výsledkem práce studentů je, že na konci hodiny se zavedou nové pojmy týkající se proměnných a datových typů.</p>
<i>Hodnocení</i>	Gamifikace představuje nejen neformální hodnocení, ale zvýší zájem, vnitřní motivaci a vzdělávací výstupy celé skupiny.
<i>Šíření výsledků</i>	Na šíření výsledků mezi učiteli a studenty se použije repozitář Github/Gitlab a systém řízení výuky (např. Moodle). Studenti mohou pokračovat v diskusi na dané téma na fóru, které je jim poskytnuté prostřednictvím nástroje na řízení výuky.

### 6.1.1. Příručka učitele na přípravu vyučovací hodiny

#### 1. Úvod (10 min)

V úvodní části je uvedený kontext související s předchozími hodinami. Učitel zavádí pojem **proměnná**.

#### 2. Identifikace proměnné (5 min)

##### Cíl:

Identifikace proměnných prostřednictvím diskuse s důrazem na úlohu proměnných v programování.

##### Pojmy na diskusi:

- proměnné a hodnoty.

##### Aktivity:

- Učitel zavádí pojem **proměnná**.
- Studenti mohou být požádáni, aby přezkoumali a určili proměnné pro svoji hru.
- O proměnných může diskutovat učitel a studenti.
- V tomto scénáři je možné typ proměnné vynechat (resp. O něm diskutovat všeobecně).

### 3. Typy údajů (15 min)

#### Cíl:

Pochopení koncepce datových typů, zjištění jejich reálných aplikací a zaměření se na typy proměnných potřebných pro vývoj her.

#### Pojmy na diskusi:

- proměnné, Datové typy, Příklady datových typů v reálném světě, Typy proměnných pro hru.

#### Aktivity:

- Učitel zavede pojem datový typ.
- Může se provést diskuse o příkladech z reálného světa (např. celá čísla mohou souviset s počtem aktuálně přítomných studentů, desetinná čísla mohou souviset s cenou výrobku, textový typ může souviset se jménem studentů atd.).
- Datové typy se posuzují v kontextu prostředí Greenfoot a programovacího jazyka Java.
- Podrobná diskuse týkající se typu proměnných potřebných pro hru.

### 4. Deklarace proměnných (10 min)

#### Cíl:

Aplikovat koncepty datových typů a proměnných a deklarovat proměnné potřebné pro vývoj her.

#### Pojmy na diskusi:

- proměnné, datové typy, typy proměnných pro hru.

#### Aktivity:

- Datové typy jsou brány v kontextu prostředí Greenfoot a programovacího jazyka Java.
- Učitel by měl vysvětlit rozdíl mezi deklarací a inicializací proměnných:
  - Při deklaraci proměnné se deklaruje proměnná určitého datového typu, ale hodnota buď může nebo nemusí být přítomna.
  - Je možné zavést analogii (např. označený box pro konkrétní typ cookie, ale bez cookie ve vnitřním označeného boxu).
- Deklarace proměnných vyžadovaných ve hře.
- Je možné zvážit další příklady. Například, pokud se uvažuje o metodě **act()**, je možné deklarovat proměnnou na zobrazení textu.

### 5. Inicializace proměnných (5 min)

#### Cíl:

Aplikovat pojmy datových typů a proměnných a inicializovat proměnné potřebné pro hru.

#### Pojmy na diskusi:

- proměnné, datové typy, hodnoty proměnných pro hru.

#### Aktivity:

- Na základě datových typů představených dříve se představí jejich hodnoty a rozsahy.
- Hodnoty údajů a rozsahy údajů se posuzují v kontextu prostředí Greenfoot a programovacího jazyka Java.

- Učitel by měl vysvětlit rozdíl mezi deklarací a inicializací proměnných:
  - Při inicializaci proměnné se deklaruje proměnná určitého datového typu a zároveň se může inicializovat (jde o tzv. počáteční hodnotu). V následujícím kódu se může její hodnota změnit.
  - Možno zavést analogii (např. cookie konkrétního typu se umístí do předem definovaného označeného pole).
- Inicializace proměnných vyžadovaných ve hře.
- Je možné zvážit další příklady. Například, pokud se uvažuje o metodě `act()`, je možné inicializovat proměnnou na zobrazení textu.

## 6.2. Úvod do operátorů a výrazů v prostředí Greenfoot

Tabulka 10. Úvod do operátorů a výrazů v prostředí Greenfoot

<i>Název</i>	Úvod do operátorů a výrazů v prostředí Greenfoot
<i>Cíle vzdělávání</i>	Na konci této kapitoly budou studenti schopni pochopit koncept operátorů. V této části se představí různé typy operátorů (tj. aritmetické operátory, logické operátory, relační operátory), stejně tak jako jejich příslušné výrazy. Kromě toho se představí objektový výraz a referenční proměnná. Probrané pojmy se budou probírat v kontextu vývoje hry, čímž se podpoří kreativita, týmová práce a nadšený přístup k programování v prostředí Greenfoot.
<i>Cílová skupina</i>	Studenti středních škol na kurzu OOP4Fun. Základní znalosti programování včetně konceptů iterace a výběru. Studenti by měli být seznámeni s programem Greenfoot.
<i>Trvání scénáře</i>	<ol style="list-style-type: none"> <li>1. Operátory (15 min) .....</li> <li>2. Aritmetické operátory a výrazy (10 min) .....</li> <li>3. Logické operátory (15 min) .....</li> <li>4. Relační operátory (10 min) .....</li> <li>5. Logické výrazy (10 min).....</li> <li>6. Objektové výrazy (5 min) .....</li> <li>7. Referenční proměnná a její hodnota null (15 min).....</li> <li>8. Úloha - Otáčení ve směru (15 min) .....</li> </ol>
<i>Materiály a zdroje</i>	<ul style="list-style-type: none"> <li>• Učebnice z projektu OOP4Fun.</li> <li>• Zdroje z projektu OOP4Fun.</li> <li>• Zdrojový kód projektu z Github/Gitlab.</li> <li>• Internetové zdroje.</li> </ul>
<i>Popis</i>	<p>V tomto 95-minutovém výukovém scénáři se studenti středních škol seznámí s programovacími koncepty, které se týkají operátorů a výrazů v kontextu vývoje her ve vývojovém prostředí Greenfoot. Hodina začne 15-minutovým úvodem učitele, který poskytuje základní informace z předchozích hodin, což vytvoří základ pro zavedení a definování operátorů.</p> <p>Potom následuje 10-minutový scénář, během kterého se proberou aritmetické operátory a výrazy. Aritmetické operátory se používají na aritmetické operace. V této souvislosti se prakticky vysvětlují a diskutují různé operátory a výrazy v prostředí Greenfoot.</p> <p>V další 15-minutové části se představí logické operátory. Jedná se o logické operátory</p>

	<p>používané na manipulaci s logickými hodnotami. Následuje 10-minutová sekce týkající se relačních operátorů, které se používají na porovnávání hodnot. Na základě předchozích relací se v dalším 10-minutovém scénáři proberou logické výrazy v kontextu prostředí Greenfoot.</p> <p>Následující 5-minutová část se zaměřuje na vyjádření objektu. Dále následuje 15-minutová část, která se zabývá referenčními proměnnými.</p> <p>Nakonec následuje 15-minutové cvičení pod vedením učitele, během kterého žáci řeší úlohy související s otáčením směru instancí v jejich hře. Součástí této části je také zpětná vazba poskytnutá učitelem a spolužáky.</p> <p>Žáci pokračují v práci na projektu hry, kterou začali tvořit dříve. Následně se seznámí s pojmy, které se týkají operátorů a výrazů.</p>
<i>Hodnocení</i>	<p>Gamifikace představuje nejen neformální hodnocení, ale také zvýší zájem, vnitřní motivaci a výsledky vzdělávání celé skupiny.</p> <p>Stav projektu otevírá možnosti pro domácí úlohy. V této souvislosti je možné zavést další třídy, výrazy a hodnoty pro dosažení dalšího chování hry. Tyto koncepty je možné se studenty prodiskutovat a příslušnou realizaci zadat jako domácí úlohu.</p>
<i>Šíření výsledků</i>	<p>Na šíření výsledků mezi učiteli a studenty se použije repozitář Github/Gitlab a systém řízení výuky (např. Moodle). Studenti mohou pokračovat v diskusi na dané téma na fóru, které je jim poskytnuto prostřednictvím nástroje na řízení výuky.</p>

### 6.2.1. Příručka učitele na přípravu vyučovací hodiny

#### 1. Operátory (15 min)

##### Cíl:

Pochopení koncepce operátorů, jejich propojení s reálnými případy a seznámení se s různými typy operátorů.

##### Pojmy na diskusi:

- proměnné, Datové typy, Operátory, Příklady operátorů z reálného světa.

##### Aktivity:

Učitel zavede pojem **operátor**. Učitel by měl tento pojem přiblížit žákům pomocí příkladů z reálného života (např. nákup produktů na trhu). Potom učitel představí různé typy operátorů.

#### 2. Aritmetické operátory a výrazy (10 min)

##### Cíl:

Pochopení konceptu aritmetických operátorů, jejich propojení s reálnými příklady a seznámení se s různými aritmetickými operátory.

##### Pojmy na diskusi:

- proměnné, datové typy, operátory, příklady aritmetických operátorů z reálného světa.

##### Aktivity:



Učitel může vysvětlit operátory, které jsou známé z jiných předmětů (např. matematické a aritmetické operátory). Tyto operátory se rozebírají v kontextu prostředí Greenfoot a programovacího jazyka Java. Učitel probere různé pojmy: operátor, operand a prioritu operátoru. Další příklady mohou zahrnovat definování lokálních proměnných na získání polohy  $x$  a  $y$  instance objektu a manipulaci s ní, čímž se změní její poloha změnou hodnot proměnných.

### 3. Logické operátory (15 min)

#### Cíl:

Pochopení konceptu logických operátorů, jejich propojení s reálnými případy a seznámení se s různými logickými operátory.

#### Pojmy na diskusi:

- proměnné, Datové typy, Operátory, Příklady booleovských operátorů z reálného světa.

#### Aktivity:

Učitel může vysvětlit operátory, které jsou známé z jiných předmětů (např. matematické a logické operátory). Tyto operátory se probírají v kontextu prostředí Greenfoot a programovacího jazyka Java. Učitel probere různé pojmy: operátor, operand a prioritu operátoru. Další příklady mohou zahrnovat definování lokálních proměnných na kontrolu, zda se vodorovná poloha  $x$  instance rovná její svislé poloze  $y$ , použití logického operátoru na určení, zda se instance nachází na diagonále.

### 4. Relační operátory (10 min)

#### Cíl:

Pochopení konceptu relačních operátorů, jejich propojení s reálnými případy a seznámení se s různými relačními operátory.

#### Pojmy na diskusi:

- proměnné, Datové typy, Operátory, Příklady operátorů z reálného světa.

#### Aktivity:

Učitel může vysvětlit operátory, které už pozná z jiných předmětů (např. matematické relační operátory). Tyto operátory se probírají v kontextu prostředí Greenfoot a programovacího jazyka Java. Učitel probere různé pojmy: operátor, operand a priorita operátoru. Další příklady mohou zahrnovat definování lokálních proměnných na kontrolu, zda je pozice  $y$  jedné instance pod  $y$  pozicí jiné instance, použití relačních operátorů na určení pozičních vztahů mezi instancemi.

### 5. Logické výrazy (10 min)

#### Cíl:

Pochopení konceptu logických výrazů, jejich propojení s reálnými případy a seznámení se s různými logickými výrazy.

#### Pojmy na diskusi:

- proměnné, Datové typy, Operátory, Příklady logických výrazů z reálného světa.

### Aktivita:

Učitel dokáže vysvětlit logické výrazy v kontextu s operátory prezentovanými dříve. Tyto výrazy se probírají v kontextu prostředí Greenfoot a programovacího jazyka Java. Učitel probere operátor, operand a prioritu operátoru v kontextu booleovských výrazů. Může se zamyslet nad dalšími příklady. Například, logické výrazy je možné použít na ověření, zda se pozice instance nachází uvnitř definovaného rozměru světa v hře.

#### 6. Objektové výrazy (5 min)

##### Cíl:

Pochopení konceptu objektových výrazů, jejich propojení s reálnými případy a seznámení se s různými objektovými výrazy.

##### Pojmy na diskusi:

- proměnné, Datové typy, Operátory, Příklady objektových výrazů z reálného světa.

##### Aktivita:

Učitel dokáže vysvětlit vyjádření objektu v kontextu objektově orientovaného návrhu. Tyto výrazy se rozebírají v kontextu prostředí Greenfoot a programovacího jazyka Java. Učitel probere operátor, operand, prioritu operátorů a přetypování objektů v kontextu s objektovými výrazy. Je také možné prozkoumat další příklady. Například porovnat reference na dvě objektové instance a zkontrolovat, zda se překrývají.

#### 7. Referenční proměnná a její hodnota null (15 min)

##### Cíl

Pochopení konceptu referenčních proměnných, jejich propojení s reálnými případy a jejich aplikace při vývoji hry.

##### Pojmy na diskusi:

- proměnné, Datové typy, Operátory, Příklady referenčních proměnných z reálného světa.

##### Aktivita:

Učitel dokáže vysvětlit referenční proměnné v kontextu objektově orientovaného návrhu. Tyto referenční proměnné se probírají v kontextu prostředí Greenfoot a programovacího jazyka Java. Učitel by měl vysvětlit referenční hodnotu **null**.

#### 8. Úloha - Otáčení ve směru (15 min)

##### Cíl:

- pochopení konceptu proměnných, datových typů, operátorů, výrazů a jejich využití při vývoji hry.

##### Pojmy na diskusi:

Proměnné, datové typy, operátory, výrazy.

##### Aktivita:

Učitel by měl prodiskutovat metodu **Enemy.act()**. Učitel vysvětlí, jakým způsobem používat lokální proměnné v kódu, například jak použít proměnnou „**rotation**“. Učitel by měl popsat rozdíl mezi **this.rotation** a **rotation**. Učitel by měl popsat chování metody **getOneIntersectingObject(\_cls\_)**. Její návratovou hodnotou je instance, ta se ukládá do správně deklarované lokální proměnné a vyžaduje se

přetypování objektu. Pokud neexistuje žádný objekt protínající se s objektem, vrátí hodnotu **null**. Na základě ohodnocení logického výrazu se vykoná příslušná činnost. O výsledcích diskutuje učitel se studenty.

Commit: [97dddc4beba40ac785c7413bb245ba849cd956d2](https://github.com/4FUN/4FUN/commit/97dddc4beba40ac785c7413bb245ba849cd956d2).

### 6.3. Úvod do konstruktorů v prostředí Greenfoot

Tabulka 11. Úvod do konstruktorů v prostředí Greenfoot

<i>Název</i>	Úvod do konstruktorů v prostředí Greenfoot
<i>Cíle vzdělávání</i>	Na konci této kapitoly budou studenti schopní porozumět konceptu konstruktorů. Na hodině se představí základní teoretické pojmy týkající se konstruktorů, stejně tak jako různé vysvětlení kódu a úlohy. Probrané pojmy se budou probírat v kontextu vývoje hry, čímž se podpoří kreativita, týmová práce a nadšený přístup k programování v prostředí Greenfoot.
<i>Cílová skupina</i>	Studenti středních škol na kurzu OOP4Fun. Základní znalosti programování včetně konceptů iterace a výběru. Studenti by měli být seznámeni s programem Greenfoot.
<i>Trvání scénáře</i>	<ol style="list-style-type: none"> <li>1. Základní principy konstruktorů (10 min) .....</li> <li>2. Vysvětlení kódu (20 min).....</li> <li>3. Úloha 4.2 - Přejmenování třídy MyWorld na Arena (5 min).....</li> <li>4. Úloha 4.3 - Vytvoření uspořádání arény (30 min) .....</li> </ol>
<i>Materiály a zdroje</i>	<ul style="list-style-type: none"> <li>● Učebnice z projektu OOP4Fun.</li> <li>● Zdroje z projektu OOP4Fun.</li> <li>● Zdrojový kód projektu z Github/Gitlab.</li> <li>● Internetové zdroje.</li> </ul>
<i>Popis</i>	<p>V průběhu tohoto 65-minutového výukového scénáře se studenti středních škol seznámí s pojmy, které se týkají konstruktoru v kontextu vývoje her pomocí nástroje Greenfoot. Vyučování začíná 10-minutovým úvodem učitele, který poskytuje základné teoretické pojmy o konstruktorech.</p> <p>Potom následuje 20-minutový scénář, během kterého se prakticky vysvětlují a diskutují různé příklady kódu v prostředí Greenfoot.</p> <p>Další 5-minutová část relace se týká úlohy: Dříve definovaná třída <b>MyWorld</b> by se měla přejmenovat. V této souvislosti by se měl definovat nový název, tj. název třídy by měl být <b>Arena</b>. Důležité je poznamenat, že konstruktor třídy by se měl také přeměňovat.</p> <p>Nakonec následuje 30-minutové cvičení pod vedením učitele, během kterého žáci řeší úlohu týkající se rozložení třídy <b>Arena</b>. Součástí tohoto cvičení je zpětná vazba poskytnutá učitelem a ostatními studenty.</p> <p>Studenti budou pokračovat v práci na herním projektu, který začali dříve. Následně se do konce hodiny seznámí s pojmy, které souvisí s konstruktory.</p>
<i>Hodnocení</i>	Gamifikace představuje nejen neformální hodnocení, ale také zvýší zájem, vnitřní motivaci a výsledky vzdělávání celé skupiny.

	<p>Stav projektu otevírá možnosti pro domácí úlohy. V této souvislosti je možné zavést další třídy, výrazy a hodnoty na dosáhnout dalšího chování hry. Tyto koncepty je možné se studenty prodiskutovat a příslušnou realizaci zadat jako domácí úlohu.</p>
<p><i>Šíření výsledků</i></p>	<p>Na šíření výsledků mezi učitelem a studenty se použije repozitář Github/Gitlab a systém řízení výuky (např. Moodle). Studenti mohou pokračovat v diskusi na dané téma na fóru, které je jim poskytnuté prostřednictvím nástroje na řízení výuky.</p>

### 6.3.1. Příručka učitele na přípravu vyučovací hodiny

#### 1. Základní principy konstruktorů (10 min)

**Cíl:**

Pochopení konceptu konstruktorů.

**Pojmy na diskusi:**

- konstruktory, třídy, objekty, klíčová slova: **super, new, this**.

**Aktivity:**

Učitel představí pojem konstruktor v kontextu pojmů třída a objekt v objektově orientovaném programování. Konstruktory se používají na inicializaci konkrétní instance třídy.

#### 2. Vysvětlení kódu (20 min)

**Cíl:**

Pochopení konceptu konstruktorů, jejich propojení s reálnými situacemi.

**Pojmy na diskusi:**

- konstruktory, třídy, objekty, metody, parametry, klíčová slova: **super, new, this**, příklady konstruktorů z reálného světa.

**Aktivity:**

Učitel by měl se studenty diskutovat o konstruktorech v kontextu konceptu třídy a konceptů OOP. Konstruktory se používají na inicializaci instancí třídy. Kromě toho jsou konstruktory vždy vyvolané a mohou být definované buď implicitně, nebo explicitně. Existují předvolené konstruktory (které jsou implicitně definované), stejně tak jako parametrické a bezparametrické konstruktory (které jsou explicitně definované programátorem). Rozdíly mezi parametrickými a bezparametrickými konstruktory by se měly také prodiskutovat. Pro lepší přiblížení pojmu konstruktor studentům by měl učitel používat příklady z reálného života.

#### 3. Úloha 4.2 - Přejmenování třídy MyWorld na Arena (5 min)

**Cíl:**

Přeměňování třídy **MyWorld**.

**Pojmy na diskusi:**

- konstruktory, třídy, objekty.

**Aktivity:**

Dříve definovaná třída **MyWorld** by se měla přeměňovat. V této souvislosti by se měl zvolit nový název, konkrétně **Arena**. Proto by se měl přejmenovat i konstruktor třídy z **MyWorld()** na **Arena()**.

**Commit:** [aaf73c9bfd9f76a2a1e504f5e78d2976f1cada12](#)

#### 4. Úloha 4.3 - Vytvoření uspořádání arény (30 min)

##### Cíl:

Pochopení konceptu konstruktorů a jejich propojení se situací ve hře.

##### Pojmy na diskusi:

- konstruktory, třídy, objekty, metody, parametry, klíčová slova: **super**, **new**, **this**.

##### Aktivity:

V této aktivitě by se mělo vytvořit vlastní rozložení pro **Arénu**. Vlastní rozložení by se mělo provést v rámci konstruktoru třídy **Arena**. Měla by se přidat jedna instance třídy **Enemy**, jedna instance třídy **Orb** a aspoň jedna instance třídy **Direction**. Po deklarování a inicializaci proměnných pro jednotlivé objekty by se těmto objektům měli přiřadit vlastnosti zavoláním jejich příslušných metod. Nakonec by se tyto objekty měli začlenit do arény zavoláním metody **addObject(Actor)**.

**Commit:** [8b105ea2eaf697f08c321efe687ddd31e2d0a041](#)

## 6.4. Úvod do atributů v prostředí Greenfoot

Tabulka 12. Úvod do atributů v prostředí Greenfoot

<i>Název</i>	Úvod do atributů v prostředí Greenfoot
<i>Cíle vzdělávání</i>	Na konci této kapitoly budou studenti schopní porozumět konceptu atributů. Na hodině se představí základní teoretické pojmy týkající se atributů, stejně tak jako různé vysvětlení kódu a úlohy. Zkoumané pojmy se budou probírat v kontextu vývoje hry, čímž se podpoří kreativita, týmová práce a nadšený přístup k programování v prostředí Greenfoot.
<i>Cílová skupina</i>	Studenti středních škol na kurzu OOP4Fun. Základní znalosti programování včetně konceptů iterace a výběru. Studenti by měli být seznámeni s programem Greenfoot.
<i>Trvání scénáře</i>	<ol style="list-style-type: none"> <li>1. Úloha 4.4 - Identifikace problému s pohybem a návrh řešení (30 min) .....</li> <li>2. Atributy (10 min) .....</li> <li>3. Parametry konstruktoru (10 min).....</li> <li>4. Úloha 4.5 - Atribut <code>moveDelay</code> třídy <code>Enemy</code> (20 min).....</li> <li>5. Úloha 4.6 - Pohyb nepřátel s dodržáním zpoždění (30 min).....</li> </ol>
<i>Materiály a zdroje</i>	<ul style="list-style-type: none"> <li>• Učebnice z projektu OOP4Fun.</li> <li>• Zdroje z projektu OOP4Fun.</li> <li>• Zdrojový kód projektu z Github/Gitlab.</li> <li>• Internetové zdroje.</li> </ul>
<i>Popis</i>	V průběhu tohoto 100-minutového výukového scénáře se studenti středních škol seznámí s pojmy, které se týkají atributů v kontextu vývoje her s využitím nástroje Greenfoot. Hodina začne 30-minutovou úlohou týkající se problémů souvisejících s pohybem a jejich možných

	<p>řešení.</p> <p>Na základě předchozí úlohy se v další 10-minutové části představí pojmy související s atributy. Následuje 10-minutový scénář, během kterého se vysvětlí a parametry konstruktorů.</p> <p>Další 20-minutové cvičení pod vedením učitele souvisí s úlohou. Definuje se nový atribut týkající se pohybu ve třídě <b>Enemy</b>. Navíc se definuje parametrický konstruktor. Součástí této části je zpětná vazba poskytnutá učitelem a spolužáky.</p> <p>Nakonec se v poslední 30-minutové části pod vedením učitele implementuje pohyb nepřátel. V této souvislosti se aktualizuje metoda <b>act()</b>. V této fázi je zahrnutá vazba poskytnutá učitelem a spolužáky.</p> <p>Studenti budou pokračovat v práci na herním projekt, který začali tvořit dříve. Následně se do konce hodiny seznámí s pojmy souvisejícími s atributy.</p>
<i>Hodnocení</i>	<p>Gamifikace představuje neformální nejen hodnocení, ale navíc zvýší zájem, vnitřní motivaci a výsledky vzdělávání celé skupiny.</p> <p>Stav projektu otevírá možnosti pro domácí úlohy. V této souvislosti je možné zavést další třídy, výrazy a hodnoty na dosažení dalšího chování hry. Tyto koncepty je možné se studenty prodiskutovat a příslušnou realizaci zadat jako domácí úlohu.</p>
<i>Šíření výsledků</i>	<p>Na šíření výsledků mezi učitelem a studenty se použije repozitář Github/Gitlab a systém řízení výuky (např. Moodle). Studenti mohou pokračovat v diskusi na dané téma na fóru, které je jim poskytnuté prostřednictvím nástroje na řízení výuky.</p>

#### 6.4.1. Příručka učitele na přípravu vyučovací hodiny

##### 1. Úloha 4.4 - Identifikace problému s pohybem a návrh řešení (30 min)

###### Cíl:

Pochopení problémů souvisejícího s pohybem a jejich možné řešení v kontextu konstruktorů a atributů.

###### Pojmy na diskusi:

- konstruktory, atributy, metody.

###### Aktivita:

Učitel by měl vysvětlit, že nepřítel (instance třídy **Enemy**) se v současnosti pohybuje najednou před dvě buňky, což způsobuje problémy s jeho pohybem. Na vyřešení tohoto problému lze rychlost nepřítel modelovat jinak. Instance třídy **Enemy** se od teď bude vždy pohybovat po jedné buňce. Navíc je možné definovat nový atribut s názvem **moveDelay**, který způsobí, že se instance třídy **Enemy** začne pohybovat až po uplynutí určitého počtu volání metody **act()**.

##### 2. Atributy (10 min)

###### Cíl:

Pochopení konceptu atributů.

###### Pojmy na diskusi:

- konstruktory, třídy, objekty, atributy.

**Aktivita:**

Učitel představí koncept atributů v kontextu konceptů třídy a objektů v objektově orientovaném programování.

3. Parametry konstruktora (10 min)

**Cíl:**

Pochopení konceptu parametrů konstruktora.

**Pojmy na diskusi:**

- konstruktory, třídy, objekty, atributy, parametry, klíčová slova: **super, new, this**.

**Aktivita:**

Učitel uvede pojem parametr konstruktora v kontextu pojmů třída a objekt v objektově orientovaném programování.

4. Úloha 4.5 - Atribut `moveDelay` třídy `Enemy` (20 min)

**Cíl:**

Pochopení pojmu atributu a parametru konstruktora.

**Pojmy na diskusi:**

- konstruktory, třídy, objekty, atributy, parametry, klíčová slova: **super, new, this**.

**Aktivita:**

Do třídy `Enemy` se přidá nový atribut s názvem `moveDelay` typu `int`. Stejně tak bude definovaný parametrický konstruktorem na inicializaci tohoto atributu, přičemž atribut bude nastavený na hodnotu uvedenou v parametru. Kód ve třídě `Arena` se upraví dle následujícího commitu.

**Commit:** [6092489ce57541e77ae4e2ee886b20853df9f8a4](https://github.com/6092489ce57541e77ae4e2ee886b20853df9f8a4).

5. Úloha 4.6 - Pohyb nepřátel s dodržáním zpoždění (30 min)

**Cíl:**

Pochopení pojmu atributu a parametru konstruktora.

**Pojmy na diskusi:**

- konstruktory, třídy, objekty, atributy, parametry, klíčová slova: `super, new, this`.

**Aktivita:**

Metoda `act()` třídy `Enemy` bude aktualizovaná tak, aby se instance třídy `Enemy` pohnula až po `moveDelay` počtu volání této metody. Navíc se zavede nový atribut s názvem `nextMoveCounter` typu `int`, který se v konstruktoru inicializuje na hodnotu `0`. Metoda `act()` se upraví tak, aby volala `this.move(1)` jen v případě, že `nextMoveCounter` dosáhne hodnoty `0`. Po vykonání pohybu se `nextMoveCounter` nastaví na hodnotu `moveDelay`. Pokud se instance třídy `Enemy` nemůže pohnout, protože `nextMoveCounter` ještě nedosáhl hodnoty `0`, tak se `nextMoveCounter` sníží o hodnotu `1`.

**Commit:** [bf26e6ed23911ccb712fae3e243cdedff3a89a7f](https://github.com/bf26e6ed23911ccb712fae3e243cdedff3a89a7f).

## 6.5. Úvod do přetěžování konstruktorů v prostředí Greenfoot

Tabulka 13. Úvod do přetěžování konstruktorů v prostředí Greenfoot

<i>Název</i>	Úvod do přetěžování konstruktorů v prostředí Greenfoot
<i>Cíle vzdělávání</i>	Na konci této kapitoly budou studenti schopní porozumět konceptu přetěžování konstruktorů. Na hodině se představí základní nejen teoretické pojmy, které se týkají přetěžování konstruktorů, ale také různé vysvětlení kódu a úlohy. Prozkoumávané pojmy se budou probírat v kontextu vývoje hry, čímž se podpoří kreativita, týmová práce a nadšený přístup k programování v prostředí Greenfoot.
<i>Cílová skupina</i>	Studenti středních škol na kurzu OOP4Fun. Základní znalosti programování včetně konceptů iterace a výběru. Studenti by měli být seznámeni s programem Greenfoot.
<i>Trvání scénáře</i>	<ol style="list-style-type: none"> <li>1. Základní principy přetěžování konstruktorů (5 min) .....</li> <li>2. Úloha 4.7 - Parametrický konstruktor třídy Direction (25 min) .....</li> <li>3. Úloha 4.8 - Přetížení konstruktoru ve třídě Direction (25 min) .....</li> <li>4. Opakování teorie (20 min) .....</li> </ol>
<i>Materiály a zdroje</i>	<ul style="list-style-type: none"> <li>• Učebnice z projektu OOP4Fun.</li> <li>• Zdroje z projektu OOP4Fun.</li> <li>• Zdrojový kód projektu z Github/Gitlab.</li> <li>• Internetové zdroje.</li> </ul>
<i>Popis</i>	<p>V průběhu tohoto 75-minutového výukového scénáře se studenti středních škol seznámí s koncepty souvisejícími s přetěžováním konstruktorů v kontextu vývoje her pomocí nástroje Greenfoot. Vyučování začíná 5-minutovým úvodem do základních pojmů přetěžování konstruktorů.</p> <p>Další 25-minutová část pod vedením učitele se týká úlohy, ve které se definuje parametrický konstruktor ve třídě <b>Direction</b>. V další 25-minutové úloze se definuje přetížený konstruktor ve třídě <b>Direction</b>. Součástí této části je zpětná vazba poskytnutá učitelem a spolužáky.</p> <p>Nakonec se v poslední 20-minutové části pod vedením učitele vykonává teoretická revize související s pojmy, které byly probrány dříve (tj. proměnné, výrazy, operátory, konstruktory, atributy a přetěžování konstruktorů).</p>
<i>Hodnocení</i>	<p>Gamifikace představuje nejen neformální hodnocení, ale zvýší zájem, vnitřní motivaci a výsledky vzdělávání celé skupiny.</p> <p>Stav projektu otevírá možnosti pro domácí úlohy. V této souvislosti je možné zavést další třídy, výrazy a hodnoty na dosáhnutí dalšího chování hry. Tyto koncepty je možné se studenty prodiskutovat a příslušnou realizaci zadat jako domácí úlohu.</p>
<i>Šíření výsledků</i>	Na šíření výsledků mezi učitelem a studenty se použije repozitář Github/Gitlab a systém řízení výuky (např. Moodle). Studenti mohou pokračovat v diskusi na dané téma na fóru, které je jim poskytnuté prostřednictvím nástroje na řízení výuky.



## 6.5.1. Příručka učitele na přípravu vyučovací hodiny

### 1. Základní principy přetěžování konstruktorů (5 min)

#### Cíl:

Pochopení principů přetěžování konstruktorů.

#### Pojmy na diskusi:

- konstruktory

#### Aktivity:

Probírají se koncepty přetěžování konstruktorů.

### 2. Úloha 4.7 - Parametrický konstruktor třídy **Direction** (25 min)

#### Cíl:

Definice parametrického konstruktoru třídy **Direction** v kontextu vývoje hry.

#### Pojmy na diskusi:

- konstruktory, parametry, atributy, parametrické konstruktory.

#### Aktivity:

V této části je pro třídu **Direction** definovaný parametrický konstruktor s jediným parametrem **rotation** typu **int**. V těle konstruktoru by se vytvořená instance měla otočit na základě hodnoty tohoto parametru. Kód ve třídě **Arena** by se měl odpovídajícím způsobem aktualizovat.

**Commit:** [3c4b9ef57ab17bac2a0abc7fc5e76ea4b6e27e4b](https://github.com/4FUN/4FUN/commit/3c4b9ef57ab17bac2a0abc7fc5e76ea4b6e27e4b).

### 3. Úloha 4.8 - Přetížení konstruktoru ve třídě **Direction** (25 min)

#### Cíl:

Definice přetížení konstruktoru třídy **Direction** v kontextu vývoje hry.

#### Pojmy na diskusi:

- konstruktory, přetížení konstruktorů.

#### Aktivity:

V této části je ve třídě **Direction** definovaný přetížený konstruktor. Přidá se bezparametrický konstruktor a v jeho těle se vyvolá parametrický konstruktor s argumentem **rotation** nastaveným na **0**. Kód ve třídě **Arena** by se měl odpovídajícím způsobem aktualizovat a podle možnosti použít bezparametrickou verzi konstruktoru třídy **Direction**.

**Commit:** [1e67e67523c66acea4e93363c9a3173302f424c8](https://github.com/4FUN/4FUN/commit/1e67e67523c66acea4e93363c9a3173302f424c8).

#### 4. Opakování teorie (20 min)

##### Cíl:

Opakování teorie v souvislosti s pojmy, které byly probrány dříve.

##### Pojmy na diskusi:

- proměnné, výrazy, operátory, konstruktory, atributy, přetížení konstruktorů.

##### Aktivity:

V této části se vykoná opakování konceptů, které byly diskutovány dříve.

## 7. Asociace

V rámci tematického celku Asociace byly vytvořené čtyři metodické materiály.

### 7.1. Objekty v prostředí Greenfoot a jejich spolupráce: zkoumání metod a asociací

Tabulka 14. Objekty v prostředí Greenfoot a jejich spolupráce: zkoumání metod a asociací

<i>Název</i>	Objekty v prostředí Greenfoot a jejich spolupráce: zkoumání metod a asociací
<i>Cíle vzdělávání</i>	Na konci této kapitoly by studenti měli dobře pochopit, jakým způsobem na sebe mohou objekty navzájem reagovat. Instance třídy <b>Enemy</b> reaguje na jiné instance, zejména na instance třídy <b>Orb</b> , v prostředí Greenfoot. Studenti by měli prokázat dovednosti při vytváření a volání metod v rámci tříd jazyka Java, konkrétně implementovat a testovat metody <b>Arena.respawn(Enemy)</b> a <b>Orb.hit(Enemy)</b> . Navíc by studenti měli pochopit a efektivně spravovat atributy tříd včetně definování a používání atributů <b>Enemy.attack</b> a <b>Orb.hp</b> . Studenti by měli ovládat zapouzdření atributů v rámci třídy, což demonstrují vytvořením getterů (tj. metod, které vrátí příslušnou hodnotu atributu, resp. atributů) a setterů (tj. metod, které nastaví příslušnou hodnotu atributu, resp. atributů). Stejně tak by studenti měli pochopit důležitost zapouzdření atributů pro bezpečný a udržitelný kód. Měli by pochopit a implementovat odesílání zpráv mezi objekty a zabezpečit, aby instance tříd efektivně komunikovaly při vykonávání herních akcí. Kromě toho by studenti měli používat techniky volání metod na řešení úloh při vývoji interaktivních her, přičemž by měli efektivně využívat syntaxi a parametry potřebné na volání metod.
<i>Cílová skupina</i>	Studenti středních škol na kurzu OOP4Fun. Základní znalosti programování včetně konceptů iterace a výběru. Studenti by měli být seznámeni s programem Greenfoot.
<i>Trvání scénáře</i>	<ol style="list-style-type: none"> <li>1. Úloha 5.1 - Diskuse o tom, co by se mělo stát, pokud nepřítel dosáhne cílového místa (10 min) .....</li> <li>2. Úloha 5.2 - Diskuse o tom, jakým způsobem by se měla instance třídy Enemy reagovat na příslušné objekty pomocí zpráv při zásahu instance třídy Obr (15 min) .....</li> <li>3. Úloha 5.3 - Atributy attack třídy Enemy a HP třídy Orb (10 min).....</li> <li>4. Metoda (15 min) .....</li> <li>5. Úloha 5.4 - Získání hodnoty atributu attack třídy Enemy (5 min) .....</li> <li>6. Úloha 5.5 - Vytvoření a testování metody respawn(Enemy) třídy Arena (10 min) .....</li> <li>7. Úloha 5.6 - Vytvoření a testování metody hit(Enemy) třídy Orb (10 min) .....</li> </ol>
<i>Materiály a zdroje</i>	<ul style="list-style-type: none"> <li>• učebnice z projektu OOP4Fun.</li> </ul>

	<ul style="list-style-type: none"> <li>• zdroje z projektu OOP4Fun.</li> <li>• zdrojový kód projektu z repozitáře Github/Gitlab.</li> <li>• internetové zdroje.</li> </ul>
<p><i>Popis</i></p>	<p>V tomto 75-minutovém výukovém scénáři se studenti středních škol seznámí s technikami jako je interakce objektů, vytváření metod a práce s atributy v prostředí Greenfoot. Cílem je zlepšit pochopení studentů o tom, jakým způsobem objekty komunikují a jakým způsobem se vzájemně ovlivňují, což je klíčový aspekt objektově orientovaného programování.</p> <p>Hodina začíná 10-minutovou diskusí o souvislosti mezi objekty nepřátel a cílového místa, přičemž se diskuse zaměří na to, co by se mělo stát, když se nepřítel dostane k cílovému místu. Tím se studenti připraví na pochopení interakcí objektů v kontextu hry.</p> <p>Následně se studenti zapojí do 15-minutové úlohy, ve které budou diskutovat o tom, jakým způsobem by měla být instance třídy <b>Enemy</b> ovlivněna příslušnými objekty s využitím zpráv v okamžiku, kdy narazí na instanci třídy <b>Orb</b>. Tato diskuse zdůrazní důležitost komunikace objektů a předávání zpráv.</p> <p>Další 10-minutová úloha se zaměří na atributy <b>Enemy.attack</b> a <b>Orb.hp</b>. Studenti je budou schopni definovat, pochopí je a uvědomí si proč jsou klíčové pro řízení herních mechanismů.</p> <p>V následujících 15 minutách se studenti zaměří na metody a naučí se je vytvářet a implementovat v rámci svých tříd. Tato část umožní lepší pochopení vytváření a volání metod.</p> <p>V další 5-minutové úloze studenti budou mít za úkol vytvořit getter pro atribut <b>Enemy.attack</b>, čímžlepší své vědomosti o zapouzdření a získávání dat atributů.</p> <p>Dalších 10 minut bude věnováno vytvoření a otestování metody <b>Arena.respawn(Enemy)</b>. Studenti implementují tuto metodu na zpracování logiky opětovného vygenerování nepřátelských objektů, čímž se docílí pochopení funkčnosti a testování metod.</p> <p>Po této úloze bude následovat poslední 10-minutová úloha, která bude zahrnovat vytvoření a testování metody <b>Orb.hit(Enemy)</b>, která bude zpracovávat logiku hry v okamžiku, kdy nepřítel zasáhne instanci třídy <b>orb</b>.</p> <p>V průběhu celého výukového scénáře budou studenti pracovat samostatně nebo v malých skupinách, čímž se podpoří spolupráce a vzájemné učení. Aktivní účastí na diskusích, kódovacích úlohách a testování metod si studenti rozvinou schopnosti kritického myšlení a řešení výpočtových problémů.</p> <p>Na závěr studenti získají komplexní znalosti o interakcích objektů, tvorbě metod a správě atributů v prostředí Greenfoot. Tyto základní dovednosti jsou potřebné pro další vývoj her. Navíc zvýší jejich celkovou zdatnost v programování.</p>
<p><i>Hodnocení</i></p>	<p>Gamifikace představuje nejen neformální hodnocení, ale také zvýší zájem, vnitřní motivaci a vzdělávací výstupy celé skupiny.</p>
<p><i>Šíření výsledků</i></p>	<p>Na šíření výsledků mezi učitelem a studenty se použije repozitář Github/Gitlab a systém řízení výuky (např. Moodle). Studenti mohou pokračovat v diskusi na dané téma na fóru, které je jim poskytnuté prostřednictvím nástroje na řízení výuky.</p>

### 7.1.1. Příručka učitele na přípravu vyučovací hodiny

#### 1. Úloha 5.1 - Diskuse o tom, co by se mělo stát, pokud nepřítel dosáhne cílového místa (10 min)

##### Cíl:

Studenti dokážou diskutovat o očekávaném chování hry v situaci, kdy se nepřítel dostane k cílovému místu, o dynamice hry a jejich výsledcích, například o poškození cílového místa nebo o ukončení hry.

##### Pojmy na diskusi:

- poškození cílového místa, odstranění nepřítele, aktivování herních událostí (např. snížení zdraví, přehrávání zvukových efektů, ukončení hry).

##### Aktivity:

Hodina začne zopakováním pojmů, které byly probrány dříve, aby se učitel ujistil, že studenti mají dobrý základ pro osvojení nových vědomostí. Učitel zapojí studenty do diskuse s cílem objasnit pojmy interakce objektů, zpráv a metod v kontextu prostředí Greenfoot.

Studenti spolupracují ve skupinách a navrhnou možnosti, které mohou nastat, když nepřítel dosáhne cílové místo v jejich hře. Rozvíjejí algoritmy, jako například snižování indexu zdraví (angl. health points, dále HP) cílového místa při kontaktu s nepřítelem a rozebírají scénáře, ve kterých HP může klesnout na nulu, což vede k ukončení hry. Případně, pokud HP cílového místa zůstane nad nulou, dohodnou se na opětovném vynoření nepřítele na jiném místě arény. Také zvažují integraci dalších herních událostí vyvolaných touto interakcí, jako jsou zvukové efekty nebo zprávy na obrazovce. V době celého sezení učitel vede diskusi a nabádá studenty, aby své navrhované algoritmy sladili s konkrétním scénářem: zabezpečit, aby se po dosažení cílového místa nepřítelem snížilo HP cílového místa, jak již bylo dříve uvedené. Toto vedení učitelem studentům pomáhá prakticky aplikovat jejich nápady, posilňuje jejich chápání herní dynamiky a interakci v rámci jejich hry.

#### 2. Úloha 5.2 - Diskuse o tom, jakým způsobem by se měla instance třídy **Enemy** reagovat na příslušné objekty pomocí zpráv při zásahu instance třídy **Orb** (15 min)

##### Cíl:

Učitel se formou diskuse se studenty zaměří na proces pochopení toho, jakým způsobem by instance třídy **Enemy** měla reagovat na jiné objekty pomocí zpráv, konkrétně v případě zásahu instance třídy **Orb**.

##### Pojmy na diskusi:

- volání metody, předávání parametrů, reference na objekt.

##### Aktivity:

Studenti spolupracují ve dvojicích, aby prozkoumali, jakým způsobem by instance třídy **Enemy** měla reagovat na instanci třídy **Orb** pomocí zpráv v rámci jejich herního scénáře. Analyzují a mapují posloupnost zpráv a akcí, které by měly nastat, pokud nepřítel zasáhne cílové místo. Cílem tohoto cvičení je prohloubit jejich pochopení volání metod, předávání parametrů a referencí na objekty v kontextu vývoje hry. Když studenti diskutují o svých nápadech a zdokonalují je, učitel je povede a usměrní tak, aby za posloupnost interakcí mezi objekty zodpovídal algoritmus, který je rozložený mezi spolupracující objekty, jak je uvedené v cílech hodiny. Pro lepší pochopení v tomto procesu může učitel zavést a použít sekvenční diagram UML na vizuální popis interakcí mezi třídami **Enemy**, **Orb**, **Arena** a **Greenfoot**. Tento vizuální nástroj studentům pomáhá lépe porozumět průběhu posílání zpráv a volání metod. Tím posilňuje pochopení probíraných

konceptů objektivě orientovaného programování a pochopení jejich aplikace v programování v jazyce Java a prostředí Greenfoot.

### 3. Úloha 5.3 - Atributy attack třídy Enemy a HP třídy Orb (10 min)

#### Cíl:

Učitel představí atributy **Enemy.attack** a **Orb.hp** a vysvětlí jejich význam při určování výsledku interakce.

#### Pojmy na diskusi:

- atributy třídy, zapouzdření.

#### Aktivita:

Učitel představí koncept atributů tříd a zapouzdření a vysvětlí, jakým způsobem mohou atributy jako **Enemy.attack** a **Orb.hp** představovat základní vlastnosti objektů ve hře. Studenti se naučí definovat a používat tyto atributy ve svém kódu na modelování síly útoku nepřítele a HP cílového místa. Začínají přidáním nového celočíselného atributu s názvem **attack** do třídy **Enemy**, přičemž do konstruktoru zahrnou parametr na inicializaci tohoto atributu. Podobně přidají do třídy **Orb** celočíselný atribut s názvem **hp** spolu s parametrickým konstruktorem na nastavení této hodnoty při vytvoření objektu. Učitel vede studenty při úpravě kódu ve třídě **Arena** tak, aby obsahoval tyto nové atributy. Tato praktická zkušenost studentům pomáhá pochopit úlohu atributů tříd při interakci objektů a důležitost zapouzdření při zachování integrity a bezpečnosti kódu.

**Commit:** [4ca1e9f25685990d2bdfe5b610c28422e0944f95](https://github.com/4FUN/4FUN/commit/4ca1e9f25685990d2bdfe5b610c28422e0944f95)

### 4. Metoda (15 min)

#### Cíl:

Učitel poskytne přehled metod a vysvětlí, jakým způsobem se používají na zapouzdření akcí a chování v rámci tříd.

#### Pojmy na diskusi:

- definice metody, volání metody, parametry, návratové hodnoty.

#### Aktivita:

Učitel začne vysvětlovat pojem metody jako zapouzdřené činnosti nebo chování v rámci třídy. Na praktických příkladech demonstruje syntaxi a strukturu definic metod. Dále ukáže, jak se metody vyvolávají na objektech. Studenti se seznámí s různými typy metod včetně těch, které pouze vykonávají akce (metody typu **void**), a těch, které vrací hodnoty (metody typu **return**). Učitel vysvětlí, jak se metodám předávají parametry, přičemž zdůrazní důležitost typů a pořadí parametrů. Prostřednictvím vedených cvičení kódování si studenti procvičí definování metod s různými typy parametrů a návratových hodnot a zavolání těchto metod na instancích objektů. Studenti dále zkoumají situace, ve kterých metody vykonávají akce, modifikují stavy objektů nebo vrací specifické hodnoty, čímž si zdokonalují svoje chápání funkčnosti metod v rámci třídy.

### 5. Úloha 5.4 - Získání hodnoty atributu attack třídy Enemy (5 min)

#### Cíl:

Učitel vysvětlí koncept metod getter a jejich účel při přístupu k hodnotám atributů.

### Pojmy na diskusi:

- přístupové metody - gettery, zapouzdření.

### Aktivita:

Učitel na začátku vysvětlí účel metod typu getter a zdůrazní, jak poskytují kontrolovaný přístup k hodnotám atributů při zachování zapouzdření. Studenti pochopí, proč je důležité používat gettery na získávání hodnot zapouzdřených atributů, čímž se posílí koncept ochrany atributů v rámci třídy. Učitel potom vede studenty procesem vytváření metody getter pro atribut **attack** ve třídě **Enemy**. Pomocí praktického přístupu studenti implementují metodu getter a zabezpečí, aby vrátila hodnotu atributu **attack**. Učitel demonstruje správnou syntaxi a strukturu definování metody getter a způsob jejího použití v rámci kódu na přístup k hodnotě atributu. Na konci aktivity by studenti měli být schopni vytvořit a použít metody getter na kontrolovaný přístup k hodnotám atributů, čímž se zlepší jejich pochopení zapouzdření a ochrany údajů v objektově orientovaném programování.

Commit: [72b7456ea4cc11416c57d72c89b6a7f7e9266e3e](https://github.com/4FUN/4FUN/commit/72b7456ea4cc11416c57d72c89b6a7f7e9266e3e)

## 6. Úloha 5.5 - Vytvoření a testování metody `respawn(Enemy)` třídy `Arena` (10 min)

### Cíl:

Učitel seznámí studenty s vytvořením a otestováním metody **`Arena.respawn(Enemy)`**, která se stará o opětovné objevení nepřátel ve hře.

### Pojmy na diskusi:

- implementace metody, testování, mechanismus hry

### Aktivita:

Učitel začne seznámením studentů s konceptem implementace metody a jejím významem při definování specifického chování instance v rámci třídy. S důrazem na praktickou aplikaci vede studenty k vytvoření metody **`respawn`** ve třídě **`Arena`**. Tato metoda, která nevrací hodnotu, přijímá jeden parametr typu **`Enemy`**. Studenti jsou usměrněni, aby v rámci této metody nastavili polohu a natočení nepřítele tak, aby odpovídala hodnotám původně nastaveným v konstruktoru. Učitel předvede správnou syntaxi a strukturu definování této metody, čímž posílí klíčové pojmy implementace metody a předávání parametrů.

Studenti následně otestují svoji metodu, aby se ujistili, že funguje správně. Vytvoří instanci tříd **`Arena`** a **`Enemy`**, ale aplikaci nespustí okamžitě. Namísto toho přesunou myší instanci třídy **`Enemy`** na nové místo, potom vstoupí do kontextového menu instance třídy **`Arena`** (kliknou pravým tlačítkem do světa na místo, kde není žádná instance) a vyvolají metodu **`respawn`**. Učitel vysvětlí, jakým způsobem zkontrolovat, aby aplikace byla pozastavena a aby bylo aktivní pole parametrů. Poté povede studenty, aby klikli pravým tlačítkem myši na instanci třídy **`Enemy`** a správně vyplnili pole parametrů. Studenti pozorují výrazy v okně parametrů a poté kliknou na tlačítko **`OK`**, aby viděli výsledek použití jejich metody **`respawn`**.

Prostřednictvím této aktivity studenti získají praktické zkušenosti se zápisem a testováním metod a pochopí, jak v rámci programování manipulovat s herními objekty. Učitel dbá na to, aby studenti pochopili každý krok, v případě potřeby jim poskytuje pomoc a krok vysvětlení, čímž posilňuje jejich porozumění implementaci metod a herních mechanik.

Commit: [43a221876b8acb4fd507175ec4c8f520121d1ab1](https://github.com/4FUN/4FUN/commit/43a221876b8acb4fd507175ec4c8f520121d1ab1)

## 7. Úloha 5.6 - Vytvoření a testování metody `hit(Enemy)` třídy `Orb` (10 min)

### Cíl:

Učitel poučí studenty o vytvoření a otestování metody **`Orb.hit(Enemy)`**, která definuje interakci v okamžiku, kdy nepřítel zasáhne cílové místo.

### Pojmy na diskusi:

- interakce metod, aktualizace stavu objektu.

### Aktivity:

Učitel na začátku vysvětlí smysl metody **`Orb.hit(Enemy)`** a zdůrazní, že obsahuje logiku interakce mezi cílovým místem a nepřítelem. Potom studenty vede k přidání této metody do třídy **`Orb`**. Tato metoda, která nevrací hodnotu, přijímá jeden parametr typu **`Enemy`**.

Při testování této metody studenti postupují krok za krokem podobně jako při testování metody **`respawn`**. Vytvoří instanci třídy **`Orb`** a instanci třídy **`Enemy`**. Bez spuštění aplikace vyvolají kontextové menu instance třídy **`Orb`** a vyberou metodu **`hit`**. Učitel se ujistí, že studenti pochopili, jak vyplnit pole parametrů po kliknutí pravým tlačítkem myši na instanci třídy **`Enemy`**, když je aplikace pozastavená. Tato akce vytvoří výraz volání metody, který potom studenti vykonají kliknutím na tlačítko OK.

Učitel zdůrazní důležitost sledování výrazu vytvořeného v okně na ověření volání metody. Toto cvičení studentům pomáhá pochopit interakci metod a proces aktualizace stavů objektů v kontextu hry. Prostřednictvím této praktické činnosti studenti získají zkušenosti s implementací a testováním metod a posilují své chápání interakce metod a chování hry. Učitel poskytuje v případě potřeby podporu a vysvětlení čímž zabezpečí, že studenti úspěšně dokončí úlohu a pochopí její význam.

Commit: [fe03d520260f172066be35055a901487bf7c2ff7](https://github.com/fe03d520260f172066be35055a901487bf7c2ff7)

## 7.2. Greenfoot objekty a jejich spolupráce: prozkoumání asociací a pokročilých volání metod

Tabulka 15. Greenfoot objekty a jejich spolupráce: prozkoumání asociací a pokročilých volání metod

<b>Název</b>	Greenfoot objekty a jejich spolupráce: prozkoumání asociací a pokročilých volání metod
<b>Cíle vzdělávání</b>	<p>Na konci této kapitoly by studenti měli více porozumět tomu, jakým způsobem mohou objekty různých tříd vytvářet asociace a efektivně se ovlivňovat v prostředí Greenfoot. Studenti by měli prokázat schopnost zavolat metodu <b><code>Orb.hit(Enemy)</code></b> ze třídy <b><code>Enemy</code></b>, čímž demonstrují svoje schopnost zavádět pokročilé volání metod a zjednodušovat komunikaci mezi objekty. Měli by být schopni vysvětlit funkčnost klíčových metod z prostředí Greenfoot, jako například <b><code>Greenfoot.stop()</code></b> a <b><code>World.getWorldOfType(_cls_)</code></b>, pochopit jejich úlohu při řízení vykonávání hry a efektivní zprávě instancí objektů a také by měli úspěšně implementovat metodu <b><code>Orb.hit(Enemy)</code></b> v rámci svých projektů a začlenit interakce objektů a funkcionality metod na vytvoření interaktivní a dynamické herní mechaniky.</p> <p>Studenti by měli aplikovat základní principy objektově orientovaného programování, včetně zapouzdření a volání metod, na vytvoření sofistikovaných herních interakcí a funkcionalit. Měli by získat praktické poznatky o různých aspektech vývoje her včetně vytváření nepřátel, řízení stavu hry a vytváření atraktivních hráčských dojmů prostřednictvím organizované interakce</p>

	objektů.
<i>Cílová skupina</i>	Studenti středních škol na kurzu OOP4Fun. Základní znalosti programování včetně konceptů iterace a výběru. Studenti by měli být seznámeni s programem Greenfoot.
<i>Trvání scénáře</i>	<ol style="list-style-type: none"> <li>1. Asociace (10 min).....</li> <li>2. Zavolání metody hit třídy Orb ze třídy Enemy (15 min).....</li> <li>3. Vysvětlení kódu metod Greenfoot.stop a World.getWorldOfType (15 min).....</li> <li>4. Úloha 5.8 - Implementace metody hit třídy Orb (30 min) .....</li> </ol>
<i>Materiály a zdroje</i>	<ul style="list-style-type: none"> <li>• učebnice z projektu OOP4Fun.</li> <li>• zdroje z projektu OOP4Fun.</li> <li>• zdrojový kód projektu z repozitáře Github/Gitlab.</li> <li>• internetové zdroje.</li> </ul>
<i>Popis</i>	<p>V tomto 70-minutovém výukovém scénáři se studenti středních škol podrobně seznámí s asociacemi mezi objekty a pokročilým voláním metod v prostředí Greenfoot. Cílem této lekce je prohloubit u studentů porozumění konceptů objektově orientovaného programování a zlepšit jejich schopnost implementovat složité interakce při vývoji her.</p> <p>Na začátku se uskuteční 10-minutová diskuse o asociacích mezi třídami se zaměřením na to, jakým způsobem mohou objekty v rámci softwarového systému vzájemně komunikovat a spolupracovat. Pochopení tohoto je základem pro zkoumání složitějších interakcí.</p> <p>Následně se studenti zapojí do 15-minutové úlohy zaměřené na volání metody <b>Orb.hit(Enemy)</b> ze třídy <b>Enemy</b>. Tato úloha klade důraz na praktické využití volání metod a předávání zpráv mezi objekty.</p> <p>Další část obsahuje podrobné vysvětlení kódu metod <b>Greenfoot.stop()</b> a <b>World.getWorldOfType(_cls_)</b>, což zabere 15 minut. Studenti získají přehled o tom, jak tyto metody fungují v rámci prostředí Greenfoot a jak umožňují přesné ovládání herních prvků a správu světa.</p> <p>Hlavní část je věnovaná 30-minutové úloze, ve které studenti implementují metodu <b>Orb.hit(Enemy)</b>. Tato úloha je pro studenty výzvou, aby využili svoje znalosti z implementace metod, předávání parametrů a interakce objektů na vytvoření funkční herní mechaniky v rámci svých projektů.</p> <p>Na konci se studenti naučí lépe chápat asociace mezi objekty, ovládat pokročilé volání metod, jako například <b>Orb.hit(Enemy)</b> ze třídy <b>Enemy</b>, a získají přehled o implementaci klíčových metod prostředí Greenfoot. Tyto dovednosti je připraví na tvorbu interaktivnějších a dynamičtějších her, přičemž budou moci naplno využít potenciál principů objektově orientovaného programování při vývoji her.</p>
<i>Hodnocení</i>	Gamifikace představuje neformální nejen hodnocení, ale také zvýší zájem, vnitřní motivaci a vzdělávací výstupy celé skupiny.
<i>Šíření výsledků</i>	Na šíření výsledků mezi učitelem a studenty se použije repozitář Github/Gitlab a systém řízení výuky (např. Moodle). Studenti mohou pokračovat v diskusi na dané téma na fóru, které je jim



	poskytnuté prostřednictvím nástroje na řízení výuky.
--	--

### 7.2.1. Příručka učitele na přípravu vyučovací hodiny

#### 1. Asociace (10 min)

##### Cíl:

Studenti porozumějí pojmu asociace mezi objekty a pochopí, jak na sebe mohou instance různých tříd vzájemně působit.

##### Pojmy na diskusi:

- asociace, interakce objektů, vztahy mezi třídami.

##### Aktivity:

Lekce začíná stručným přehledem asociací mezi třídami v objektově orientovaném programování. Učitel zapojí studenty do diskuse s cílem objasnit, jakým způsobem objekty vzájemně komunikují prostřednictvím asociací, přičemž na ilustraci těchto pojmů použije praktické příklady z prostředí Greenfoot. Studenti se více seznámí s pojmem asociace a více ho pochopí. Budou vědět, že asociace definuje, jak třídy spolupracují. Příkladem toho může být to, jakým způsobem třída **Enemy** ovlivňuje třídu **Orb** ve scénáři hry.

Prostřednictvím podrobné diskuse učitel rozebere různé typy asociací v rámci prostředí Greenfoot: jedna k jedné, jedna k vícerym a vícero k vícerym. Například asociace jedna k jedné by mohla představovat to, jak je hráč propojený se svým ztělesněním postavy, asociace jedna k vícerym by mohla představovat to, jak jeden svět obsahuje vícero instancí herců, a asociace vícero k vícerym by mohla zobrazovat to, jak různí nepřátelé reagují na více cílových míst v rámci úrovně hry.

Pomocí UML diagramů tříd určených pro Greenfoot učitel vizuálně znázorní tyto vztahy a pomůže studentům pochopit, jakým způsobem jsou asociace strukturované a implementované v jejich herních projektech. Diagram by například mohl znázorňovat, jak je třída **Enemy** spojená s vícerymi instancemi třídy **Orb**, což naznačuje vztah jednoho k vícerym, kde každý nepřítel ovlivňuje několik cílových míst.

Studenti se aktivně podílejí na rozpoznávání a určování těchto asociací v rámci svých herních projektů. Zkoumají, jakým způsobem na sebe reagují instance tříd na základě těchto vztahů a diskutují o důsledcích pro jejich herní mechaniky a logiku. Učitel uvádí konkrétní příklady z kontextu jejich vývoje hry, na kterých ilustruje, jak instance třídy **Enemy** restuje na instanci třídy **Orb** a jak se tyto interakce řídí asociacemi.

Na konci aktivity studenti získají pevnou představu o úloze, kterou asociace sehrávají při navrhování a implementaci interaktivních systémů v rámci prostředí Greenfoot. Dokáží identifikovat různé typy asociací a použít jejich znalosti na efektivní modelování a implementaci komplexních interakcí mezi objekty v prostředí Greenfoot. Toto praktické pochopení posilňuje jejich schopnost navrhovat ucelené a interaktivní herní scénáře s využitím principů objektově orientovaného programování.

#### 2. Zavolání metody hit třídy Orb ze třídy Enemy (15 min)

##### Cíl:

Učitel seznámí studenty s postupem volání metody **Orb.hit(Enemy)** ze třídy **Enemy**.

### Pojmy na diskusi:

- vyvolání metody, reference na objekty.

### Aktivita:

Studenti se zapojí do praktického učení změnou metody **act()** třídy **Enemy**. Odstraní existující kód odpovědný za již nepotřebné chování, jako je otáčení při dosažení cílového místa nebo odrážení se od okrajů světa. Namísto toho implementují funkcionalitu zavolání metody **Orb.hit(Enemy)** v okamžiku, kdy se instance třídy **Enemy** střetne s instancí třídy **Orb**.

Učitel na praktických příkladech a ukázkách ilustruje, jak správně připravit volání metody. Studenti se naučí používat referenci na danou instanci (**this**) na spuštění metody **hit()** pro instanci třídy **Orb** při střetnutí s instancí třídy **Enemy**. Dále studenti prozkoumají průběh řízení v prostředí Greenfoot a pochopí, jak volání metody určuje cestu vykonání kódu v rámci jejich herního scénáře.

V průběhu celé aktivity učitel poskytuje usměrnění na ladění a testování implementace, aby se ujistil, že volání metody **Orb.hit(Enemy)** funguje tak, jak má. Studenti pozorují chování instancí v prostředí Greenfoot a ověřují, zda volání metody účinně spouští očekávané interakce mezi instancemi tříd **Enemy** a **Orb**.

Na konci studenti získají dovednosti ve volání metod a interakci objektů v rámci programování v prostředí Greenfoot. Pochopí, jakým způsobem využívat reference na objekty na vyvolání metod v různých třídách, čímž se posilní jejich pochopení principů objektově orientovaného programování v kontextu vývoje her.

**Commit:** [63f9c96717d9d2587b60095e3b249b0158c8587b](https://github.com/63f9c96717d9d2587b60095e3b249b0158c8587b)

### 3. Vysvětlení kódu metod **Greenfoot.stop** a **World.getWorldOfType** (15 min)

#### Cíl:

Učitel vysvětlí funkčnost metody **Greenfoot.stop()** a metody **World.getWorldOfType(\_cls\_)**.

#### Pojmy na diskusi:

- metody pro kontrolu hry, správa světa.

#### Aktivita:

V rámci této vyučovací aktivity se studenti seznámí s dvěma klíčovými metodami v rámci prostředí Greenfoot:

- **Greenfoot.stop()**
- **World.getWorldOfType(\_cls\_)**.

Učitel začne objasněním účelu použití těchto metod, přičemž zdůrazní jejich role při ovládní hry a správě světa.

Metoda **Greenfoot.stop()** je pro řízení průběhu vykonávání herního scénáře v prostředí Greenfoot nezbytná. Po jejím vyvolání se hra zastaví a zmrazí se všichni aktéři a interakce ve světě. Tato metoda je obzvláště užitečná na implementaci funkce pozastavení hry nebo na spuštění specifických událostí, které si vyžadají dočasné zastavení postupu hry.

Metoda **World.getWorldOfType(\_cls\_)** má ve správě světa jiný účel. Umožňuje vývojářům získat instance světů, které jsou konkrétního typu třídy **\_cls\_**. Prochází všechny aktivní světy v prostředí Greenfoot a vrátí instanci třídy světa, která odpovídá zadanému typu. Tato možnost je výhodná v okamžiku, kdy vývojáři potřebují dynamicky komunikovat se světy nebo s nimi manipulovat na základě atributů jejich tříd.

Studentům jsou demonstrovány praktické ukázky a příklady, aby lépe pochopili tyto metody. Učitel předvede, jakým způsobem je možné integrovat metodu **Greenfoot.stop()** do herních scénářů s cílem vytvořit funkci pro pozastavení nebo spuštění specifické události ve hře. Studenti pozorují, jak pozastavení hry ovlivňuje chování aktérů a interakce v prostředí Greenfoot.

Podobně se studenti prostřednictvím praktických cvičení seznamují s metodou **World.getWorldOfType(\_cls\_)**. Naučí se, jakým způsobem tuto metodu používat na dynamické získávání instancí konkrétních typů světů. Učitel demonstruje scénáře, ve kterých je načtení světů konkrétního typu třídy potřebné na implementaci pokročilých herních mechanismů nebo na správu vícerých souběžných herních prostředí v rámci prostředí Greenfoot.

V průběhu celé aktivity učitel podporuje diskusi a poskytuje praktické příklady v programování na ilustraci použití těchto metod v reálných scénářích vývoje her. Studenti se aktivně podílejí na experimentování s metodami v rámci vlastních projektů v prostředí Greenfoot, čímž si posilují porozumění prostřednictvím přímého aplikace a zkoumání.

Na konci studenti získají dovednosti v používání funkce **Greenfoot.stop()** na ovládání hry a funkce **World.getWorldOfType(\_cls\_)** na efektivní zprávu světa v prostředí Greenfoot. Též získají praktické dovednosti, které zvýší jejich schopnost implementovat komplexní chování her a efektivně spravovat stavy hry pomocí těchto základních metod.

#### 4. Úloha 5.8 - Implementace metody hit třídy Orb (30 min)

##### Cíl:

Učitel vede studenty při implementaci metody **Orb.hit(Enemy)**.

##### Pojmy na diskusi:

- implementace metod, aktualizace stavu objektů, herní mechaniky.

##### Aktivita:

Studenti začnou implementovat metodu **Orb.hit(Enemy)**, která je klíčovým krokem při definování interakce mezi nepřítelem a cílovým místem v rámci jejich herního scénáře.

Metoda **Orb.hit(Enemy)** hraje klíčovou úlohu při určování důsledků kontaktu nepřítele s cílovým místem ve hře. Zde je vysvětlený postup, jakým způsobem mohou studenti přistupovat k této metodě a implementovat ji. Nejdříve je nutné snížit HP instance třídy **Orb**. Po zavolání metody **Orb.hit(Enemy)** by mělo dojít ke snížení HP cílového místa. Tato akce simuluje poškození, které cílové místo utrpí při kontaktu s nepřítelem. Potom je nutné zkontrolovat, zda HP cílového místa klesl na nulu. Pokud HP dosáhlo na hodnotu nula nebo méně, tak by mělo dojít k ukončení hry. To se zabezpečí vyvoláním funkce **Greenfoot.stop()**, která zastaví vykonávání hry. To je případ ukončení hry. Jiný případ nastane, pokud je HP cílového místa po střetnutí s nepřítelem stále nad nulou. Studenti by měli do hry zahrnout logiku na opětovné objevení se nepřítele zpět v aréně, aby hra mohla pokračovat.

V průběhu aktivity učitel pomáhá při implementaci metody **Orb.hit(Enemy)** tím, že poskytuje instrukce o struktuře metody, o použití parametru typu **Enemy** a způsobu aktualizace parametru HP. Studenti spolupracují, diskutují a rozhodují o konkrétních herních mechanikách, které chtějí implementovat. Například o tom, jaké poškození způsobí cílovému místu každý typ nepřítele a co se stane, když HP cílového místa klesne pod hodnotu **1**.

Učitel vyzve studenty, aby své implementace důkladně otestovali a ujistili se, že se metoda v různých herních scénářích chová podle očekávání. Na konci aktivity Studenti získají praktické zkušenosti s

implementací logiky metody na efektivní řízení herních interakcí, čímž se posílí jejich pochopení implementace metody a herní mechaniky v rámci prostředí Greenfoot.

Commit: [84bcd7c128faaa9313b507f7438f826ae2f47d2c](https://github.com/4FUN/4FUN/commit/84bcd7c128faaa9313b507f7438f826ae2f47d2c)

### 7.3. Greenfoot objekty a jejich spolupráce: Věže, střely a strategické interakce

Tabulka 16. Greenfoot objekty a jejich spolupráce: Věže, střely a strategické interakce

<b>Název</b>	Greenfoot objekty a jejich spolupráce: Věže, střely a strategické interakce
<b>Cíle vzdělávání</b>	Na konci této kapitoly získají studenti dovednosti v rámci vytváření tříd <b>Bullet</b> a <b>Tower</b> , čímž si vytvoří základ pro vývoj strategických her. Pochopí a implementují realistický pohyb střely a určí akce v případě, že se instance třídy <b>Bullet</b> střetne s instancemi třídy <b>Enemy</b> nebo s okrajem arény. Studenti navrhnu efektivní střeleckou mechaniku pro instance třídy <b>Tower</b> , přičemž využijí zasílání zpráv mezi instancemi třídy <b>Tower</b> a dalšími herními prvky na posílení herní dynamiky. Strategicky rozmístí instance třídy <b>Tower</b> v rámci herní arény a uplatní principy objektově orientovaného programování, jako je zapouzdření a volání metod, aby zabezpečili vytvoření robustní a dobře udržitelné herní mechaniky. Prostřednictvím společného řešení problémů budou studenti řešit problémy při navrhování a implementaci strategických interakcí mezi věžemi, střelami a ostatními herními prvky, získají praktické poznatky o principech herního designu a zlepší své celkové pochopení herní mechaniky v hře typu <b>Tower defense</b> v prostředí Greenfoot.
<b>Cílová skupina</b>	Studenti středních škol na kurzu OOP4Fun. Základní znalosti programování včetně konceptů iterace a výběru. Studenti by měli být seznámeni s programem Greenfoot.
<b>Trvání scénáře</b>	<ol style="list-style-type: none"> <li>1. Úloha 5.9 - Přidání tříd Bullet a Tower (10 min) .....</li> <li>2. Úloha 5.10 - Diskuse o tom, jak se má pohybovat instance třídy Bullet a co se má stát, když zasáhne instanci třídy Enemy nebo okraj arény. (10 min).....</li> <li>3. Úloha 5.11 - Implementace pohybu instance třídy Bullet (30 min).....</li> <li>4. Úloha 5.12 - Diskuse o tom, jak instance třídy Tower vystřelí instanci třídy Bullet (15 min).....</li> <li>5. Úloha 5.13 - Diskuse o tom, jak by měla instance třídy Tower komunikovat s příslušnými objekty pomocí zpráv při střelbě (15 min) .....</li> <li>6. Úloha 5.14 - Implementace střelby instance třídy Tower (30 min) .....</li> <li>7. Úloha 5.15 - Věže v aréně (20 min).....</li> </ol>
<b>Materiály a zdroje</b>	<ul style="list-style-type: none"> <li>• učebnice z projektu OOP4Fun.</li> <li>• zdroje z projektu OOP4Fun.</li> <li>• zdrojový kód projektu z repozitáře Github/Gitlab.</li> <li>• internetové zdroje.</li> </ul>
<b>Popis</b>	<p>V tomto 130-minutovém výukovém scénáři se studenti středních škol seznámí s dynamikou interakcí mezi věžemi a střelami v prostředí Greenfoot. Scénář se zaměřuje na rozvoj studentských dovedností při navrhování a implementaci strategických herních prvků s cílem vytvořit atraktivní a interaktivní hru.</p> <p>Vyučování začíná 10-minutovou úlohou, ve které studenti vytvoří třídy <b>Bullet</b> a <b>Tower</b>. Tento základní krok vytváří předpoklady pro pochopení a implementaci interakcí mezi těmito</p>

	<p>herními prvky.</p> <p>Potom následuje 10 minut diskuse o tom, jak by se měly pohybovat instance třídy <b>Bullet</b> a jaké akce by se měly vykonat, když střela dosáhne instance třídy <b>Enemy</b> nebo okraj arény. Táto diskuse je základem pro implementaci přesné a dynamické mechaniky pohybu.</p> <p>Studenti se poté věnují po dobu 30-ti minut implementaci pohybu instancí třídy <b>Bullet</b>. Tato úloha je pro ně výzvou, aby použili svoje znalosti v rámci metody <b>move()</b> v prostředí Greenfoot a znalosti ve zpracování událostí na simulaci realistického chování střely v herním prostředí.</p> <p>Pokračuje se řešením úlohy v trvání 15 minut, ve které se diskutuje o tom, jakým způsobem by měli instance třídy <b>Tower</b> střeliet instance třídy <b>Bullet</b>. Tato diskuse se zabývá logikou a podmínkami pro inicializaci střelby z věží s využitím třídy <b>Bullet</b>.</p> <p>Poté následuje dalších 15 minut věnovaných diskusi o tom, jakým způsobem by měli instance třídy <b>Tower</b> při střelbě komunikovat s příslušnými objekty při využití zpráv. Tento segment zdůrazňuje důležitost komunikace objektů a spouštění událostí při vývoji her.</p> <p>Studenti potom stráví 30 minut implementací mechanismu střelby instance třídy <b>Tower</b>. Tato úloha vyžaduje, aby studenti začlenili logiku střelby do interakce objektů a zabezpečili, aby věž efektivně zasahovali nepřítele nebo jiné herní prvky.</p> <p>Na závěr této části je 20 minutová úloha zaměřená na zprávu a rozmístování věží v aréně. Při této úloze se zkoumá umístění, interakce a strategické rozmístění věží s cílem optimalizovat dynamiku hry a vytvořit skutečné výzvy pro hráče.</p> <p>Na konci studenti získají praktické zkušenosti s navrhováním a implementací strategických interakcí mezi věžemi a střelami v prostředí Greenfoot. Navíc získají dovednosti v oblasti objektově orientovaného programování, zpracování událostí a strategického návrhu hry, což je připraví na tvorbu poutavých a dynamických her.</p>
<i>Hodnocení</i>	Gamifikace představuje neformální nejen hodnocení, ale také zvýší zájem, vnitřní motivaci a vzdělávací výstupy celé skupiny.
<i>Šíření výsledků</i>	Na šíření výsledků mezi učitelem a studenty se použije repozitář Github/Gitlab a systém řízení výuky (např. Moodle). Studenti mohou pokračovat v diskusi na dané téma na fóru, které je jim poskytnuté prostřednictvím nástroje na řízení výuky.

### 7.3.1. Příručka učitele na přípravu vyučovací hodiny

#### 1. Úloha 5.9 - Přidání tříd Bullet a Tower (10 min)

##### Cíl:

Studenti dokážou vytvořit třídy **Bullet** a **Tower**.

##### Pojmy na diskusi:

- vytvoření třídy, role třídy a výchozí nastavení.

### Aktivita:

Lekce začíná přehledem základních pojmů, které se týkají vytváření objektů, jejich pohybu a interakce v prostředí Greenfoot. Učitel zapojí studenty do diskuse s cílem objasnit úlohy různých tříd a jejich interakce ve hře.

Studenti vytvoří dvě nové třídy v prostředí Greenfoot a pochopí účel každé z nich v kontextu hry. Naučí se, jakým způsobem tyto třídy nastavit, čímž se připraví na složitější interakce v následujících lekcích.

Studenti začnou vytvořením třídy **Bullet**. Tato třída bude reprezentovat střely vystřelené věžemi. V programu Greenfoot vyberou z nabídky třídy **Actor** (kliknutí pravým tlačítkem myši) položku „**Nová podtřída**“ a pojmenují ji **Bullet**.

Potom studenti vytvoří třídu **Tower**. Tato třída bude představovat věže, které střílejí střely na nepřátele. Z nabídky třídy **Actor** opět vyberou položku „**Nová podtřída**“ a pojmenují ji **Tower**.

V průběhu celé hodiny učitel vysvětluje úlohy tříd **Bullet** a **Tower** ve hře. Třída **Bullet** představuje střely, které budou střílet věže, zatímco třída **Tower** představuje stacionární objekty, které mohou střílet střely na nepřátele. Učitel se ujistí, že studenti rozumí rozdělaným úlohám, které bude každá třída mít v rámci hry, a tomu, jakým způsobem na sebe budou ve hře vzájemně působit.

Na konci této aktivity by měli studenti vytvořit a nastavit základní struktury tříd **Bullet** a **Tower**, čím vytvoří základ pro podrobnější implementaci na dalších hodinách.

**Commit:** [ece4df70042c8f60098e14ad2cee55514897d825](https://github.com/epic42/4FUN/blob/master/commit/ece4df70042c8f60098e14ad2cee55514897d825)

2. Úloha 5.10 - Diskuse o tom, jak se má pohybovat instance třídy **Bullet** a co se má stát, když zasáhne instanci třídy **Enemy** nebo okraj arény. (10 min)

#### Cíl:

Učitel vede diskusi o očekávaném chování střely při jejím pohybu ve hře.

#### Pojmy na diskusi:

- logika pohybu, detekce kolizí.

#### Aktivita:

Studenti diskutují o logice pohybu střely a řešení kolizí.

Studenti začnou diskutovat o tom, jako by se měla instance třídy **Bullet** pohybovat v rámci hry. Měli by dospět k závěru, že střela by se měla pohybovat po přímce ve směru, kterým byla vystřelená, bez změny směru. Rychlost střely by měla být kontrolovatelná a konzistentní. Na realizaci tohoto cíle mohou použít metody pohybu zabudované v prostředí Greenfoot. Zde je důležité zdůraznit úlohu konstruktora na inicializaci hodnot atributů při vytváření instancí tříd.

Učitel se potom v diskusi zaměří na to, co by se mělo stát, když střela dosáhne okraj světa nebo když se střetne s nepřítelem. Studenti by měli dospět k závěru, že když střela dosáhne okraj světa, měla být odstraněna ze hry. Také diskutují o situaci, zda by po střetu s nepřítelem měla střela zmizet a nepřítel by měl být zraněný nebo zničený.

Studenti by potom měli navrhnout pseudokód pro logiku pohybu a kolize střely. Příkladem může být tento pseudokód:

- Nejdříve se střela pohybuje dopředu konstantní rychlostí, přičemž se kontroluje, zda střela dosáhla okraj světa.
- Pokud dosáhla okraj světa, tak odstraň střelu ze světa.
- Pokud ne, zkontroluje se, zda se střela střetla s nepřítelem.
  - Pokud došlo ke střetu, znamená to, že střela zasáhla nepřítele a nutné ji odstranit ze světa a aplikovat poškození na nepřítele nebo ho odstranit

Učitel studenty provede způsobem, jak implementovat tuto logiku pomocí metod **move(int)**, **isAtEdge()** a **getOneIntersectingObject(Class cls)** v prostředí Greenfoot.

Učitel vyzve studenty, aby svoje nápady dopracovali a přemýšleli nad dalšími detaily, například úpravou rychlosti v závislosti od obtížnosti hry nebo přidání vizuálních efektů v okamžiku, kdy střela zasáhne nepřítele. Tato aktivita studentům pomáhá pochopit principy detekce pohybu a kolizí při vývoji her a připravuje je na implementaci v jejich budoucích projektech.

### 3. Úloha 5.11 - Implementace pohybu instance třídy **Bullet** (30 min)

#### Cíl:

Učitel studentům pomáhá implementovat logiku pohybu pro třídu **Bullet**.

#### Pojmy na diskusi:

- pohyb aktéra, hranice světa.

#### Aktivita:

Studenti napíší kód na posouvání střely dopředu a na její odstranění, když dosáhne okraj světa. Studenti začnou zkoumáním konceptu pohybu aktéra a hranic světa, přičemž se zaměřují na to, jak se tyto koncepty vztahují na třídu **Bullet**. Učitel studentům připomene předchozí úlohy a zdůrazní vědomosti, které by měli použít. Studenti si připomenou, jakým způsobem přidat kód do metody **act()** na zpracování interakcí na okraji světa. Stejně tak si připomenou, jak zpracovat změny směru, když aktér vstoupí do konkrétních buněk. Navíc si vzpomenou, jakým způsobem používat počítadla a mechanismy zpoždění v metodě **act()** na řízení pohyb.

S těmito poznatky studenti pokračují v implementaci logiky pohybu pro třídu **Bullet**. Začnou přidáním metody **move(int)** do metody **act()** třídy **Bullet**, aby se střela mohla kontinuálně pohybovat dopředu.

**Commit:** [d372827a831381b2254f838041fa4d9a42e53b82](https://github.com/d372827a831381b2254f838041fa4d9a42e53b82)

### 4. Úloha 5.12 - Diskuse o tom, jak instance třídy **Tower** vystřelí instanci třídy **Bullet** (15 min)

#### Cíl:

Učitel diskutuje se studenty o logice střelby z věže.

#### Pojmy na diskusi:

- vytvoření objektu, volání metody.

#### Aktivita:

Studenti si rozmyslí, jakým způsobem věž vytvoří a vystřelí střely. Studenti začnou diskutovat o celkové logice potřebné na to, aby věž mohla ve hře vystřelovat střely. Zaměření se na klíčové pojmy, jako je

vytváření objektu střely a vyvolávání metod na její vystřelení. Učitel zdůrazní, že věž by neměla střílet střely při každém volání metody **act()**, podobně jak tomu bylo u pohybu nepřítele, který se vyřešil s využitím mechanismu zpoždění.

Studenti si formou diskuse zformulují kroky potřebné na to, aby věž vystřelovala střely s přestávkami. Učitel je vede k tomu, aby zvážili použití principu zpoždění při střelbě. Je potřebné vysvětlit, jaká by mohla být úloha konstruktora pro implementaci tohoto mechanismu.

Učitel vysvětlí, že ve třídě **Tower** budou muset vytvořit nový atribut **shootDelay** a počítadlo **nextShootCounter**. Tyto atributy budou řídit frekvenci střelby.

Učitel rozebere příslušné kroky a vhodné metody pro třídu **Tower**. Nejdříve se v třídě **Tower** definují atributy **shootDelay** a **nextShootCounter**, přičemž se inicializuje počítadlo **nextShootCounter** na hodnotu **0** a atribut **shootDelay** se nastaví na příslušnou hodnotu. Potom je potřebné upravit metodu **act()** třídy **Tower**, aby bylo možné obsluhovat logiku střelby. Metoda by měla vytvořit a vystřelit střelu až v době, když **nextShootCounter** dosáhne hodnotu **0**. Po vystřelení by se **nextShootCounter** měl vynulovat na hodnotu **shootDelay**. Pokud hodnota **nextShootCounter** není **0**, měl by se snížit o hodnotu **1**. Na konci by se měla definovat samostatná metoda **fire()**, která by se starala o vytvoření a vystřelení střely. Tato metoda vytvoří instanci objektu **Bullet** a přidá ji do světa.

Tímto postupem studenti pochopí, jak se implementuje logika střelby oddělením příslušných kroků do metod třídy **Tower**. Učitel dbá na to, aby studenti pochopili důležitost volání metod a vytváření objektů. To u nich posilňuje pochopení těchto pojmů v kontextu jejich hry.

#### 5. Úloha 5.13 - Diskuse o tom, jak by měla instance třídy **Tower** komunikovat s příslušnými objekty pomocí zpráv při střelbě (15 min)

##### Cíl:

Učitel vysvětlí, jak věž komunikuje se střelami a jinými předměty pomocí zpráv.

##### Pojmy na diskusi:

- předávání zpráv, volání metod.

##### Aktivity:

Studenti diskutují o mechanismu předávání zpráv při střelbě věže.

Na začátku hodiny učitel vysvětlí koncept předávání zpráv a jeho význam v objektově orientovaném programování. Učitel by měl zdůraznit, jakým způsobem mezi sebou objekty komunikují v průběhu hry pomocí metod, které slouží jako zprávy.

Na ilustraci může učitel použít sekvenční diagram UML, na kterém popíše interakce mezi instancemi tříd **Tower**, **Bullet** a **Arena**. Diagram vizuálně znázorňuje tok zpráv a volání metod, čímž pomáhá studentům pochopit posloupnost interakcí.

Studenti diskutují o podrobném postupu, jakým způsobem by se věž měla při střelbě interagovat se střelami a jinými předměty. Učitel vysvětlí, že když se věž rozhodne střílet, pošle zprávu (zavolá metodu) na vytvoření instance třídy **Bullet** a přidá jí do arény. Táto interakce se spustí v rámci metody **act()** třídy **Tower**. Učitel s využitím sekvenčního diagramu UML předvede, jak věž posílá zprávu do prostředí Greenfoot na přidání nové instance třídy **Bullet** do světa. Věž potom odešle zprávu instanci třídy **Bullet** a nastaví její směr tak, aby odpovídal aktuální rotaci věže. Tím se zabezpečí, že se střela bude pohybovat v určeném směru. V okamžiku, když je střela vytvořená a umístěná, bude reagovat na jiné objekty ve hře, jako jsou nepřátelé



nebo okraje světa. Učitel vysvětlí, jak se tyto interakce zpracují v rámci metody **act()** třídy **Bullet**, která může zahrnovat kontrolu kolizí a v případě potřeby odstranění střely.

Učitel zdůrazňuje kooperativní povahu těchto interakcí a ukazuje, jakým způsobem se algoritmus rozdělí mezi spolupracující objekty. To studentům pomáhá ocenit modulární design a přehledné formy komunikace v rámci instancí v rámci hry.

## 6. Úloha 5.14 - Implementace střelby instance třídy **Tower** (30 min)

### Cíl:

Učitel usměrňuje studenty při realizaci mechanismu střelby pro třídu **Tower**.

### Pojmy na diskusi:

- vytvoření objektu, umístění aktéra.

### Aktivity:

Studenti napíší kód, který umožní věži střílet.

Na začátku učitel vysvětlí celkový cíl: implementovat mechanismus střelby pro třídu **Tower**. Učitel potom rozdělí úlohu na zvládnutelné kroky a vede je při každém z nich.

Nejdříve si studenti připraví potřebné atributy a konstruktory pro třídu **Tower**. Učitel vysvětlí, že věž potřebuje atribut na sledování toho, zda může střílet. Potom studenti vytvoří dvě metody. Metodu typu boolean **Tower.canShoot()** a metodu bez návratové hodnoty **Tower.fire()**. Zpočátku může metoda **Tower.canShoot()** vrátit hodnotu **false**, resp. Nic nedělat, takže je možné ji použít v metodě **act()**. Metoda **act()** se potom aktualizuje tak, aby používala tyto metody. Učitel vysvětlí, že metoda **canShoot()** by měla vrátit **true**, pokud počítadlo **shootCounter** dosáhne hodnoty **0**. Metoda **fire()** je implementována tak, aby vytvořila instanci třídy **Bullet** a správně ji umístila.

Učitel se ujistí, že studenti rozumějí každé části kódu, přičemž klade důraz na vytvoření střely, její umístění ve světě a nastavení jejího natočení podle věže.

Studenti potom otestují svá řešení spuštěním hry, umístěním instance třídy **Tower** a ověřením, že střely střílí ve vhodných intervalech. Učitel vyzve studenty, aby odstranili případné problémy a ujistili se, že střely jsou vytvořené a pohybují se dle očekávání.

Na konci aktivity mají studenti implementovaný funkční mechanismus střelby pro věž a upevní si tak své znalosti o vytváření objektů, umisťování aktérů a volání metod v prostředí Greenfoot.

**Commit:** [62aec085954beacf996865a55bed312a09c675f2](https://github.com/62aec085954beacf996865a55bed312a09c675f2)

## 7. Úloha 5.15 - Věže v aréně (20 min)

### Cíl:

Učitel vede studenty k tomu, aby do hry zapojili věže a střely a vytvořili tak funkční arénu.

### Pojmy na diskusi:

- integrace her, testování.

### Aktivity:

Studenti umísťujú veže do arény a testujú jejich interakci se střelami a nepřáteli. Učitel na začátku vysvětlí cíl: začlenit veže do arény a zabezpečit jejich správnou interakci se střelami a nepřáteli. Součástí hodiny bude umísťování věží do arény a testování jejich chování v rámci herního prostředí.

Studenti začnou umísťováním instancí třídy **Tower** na různé pozice v aréně. Učitel vysvětlí, jakým způsobem přidávat veže prostřednictvím programu Greenfoot a jakým způsobem zabezpečit, aby každá věž byla správně umístěná.

Dále učitel představí koncept přetěžování konstruktorů. To je užitečné zejména při vytváření instancí třídy **Tower** s různými rotacemi. Učitel potom vede studenty aktualizací třídy **Tower** tak, aby obsahovala přetížený konstruktor, který přijímá celočíselný parametr pro rotaci. To umožňuje větší kontrolu nad umístěním a orientací věží v aréně.

Commit: [bfb6a271f490c341c760e654b3f86a87111c54cb](https://github.com/Erasmus/4FUN/commit/bfb6a271f490c341c760e654b3f86a87111c54cb)

## 7.4. Objekty v prostředí Greenfoot a jejich spolupráce: střely, nepřátelé a dynamika hry

Tabulka 17. Objekty v prostředí Greenfoot a jejich spolupráce: střely, nepřátelé a dynamika hry

<b>Název</b>	Objekty v prostředí Greenfoot a jejich spolupráce: střely, nepřátelé a dynamika hry
<b>Cíle vzdělávání</b>	Během této kapitoly získají studenti dovednosti v navrhování a implementaci interaktivní herní dynamiky pomocí tříd <b>Bullet</b> a <b>Enemy</b> v prostředí Greenfoot. Pochopí, jak ulehčit interakci objektů prostřednictvím předávání zpráv, což umožní efektivní komunikaci mezi herními prvky. Studenti projeví schopnost implementovat přesné mechanismy na detekci kolizí a reakce, konkrétně s podrobným popisem interakce instancí třídy <b>Bullet</b> s instancemi třídy <b>Enemy</b> . Získají praktické poznatky o základních metodách prostředí Greenfoot, kterými jsou <b>Greenfoot.showText(String, int, int)</b> , <b>Greenfoot.getRandomNumber(int)</b> a <b>World.act()</b> a využijí je na zlepšení prezentace hry, zavedení náhodnosti a správu aktualizací stavu hry. Navíc studenti zvládnou implementaci mechaniky vytváření nepřítele a podmínek ukončení hry, čímž se zabezpečí dynamický zážitek ze hry. Prostřednictvím přehodnocení asociací objektů si studenti upevní svoje pochopení toho, jak objekty spolupracují při vytváření atraktivní herní dynamiky, čímž se připraví na uplatnění těchto dovedností pro budoucí projekty v rámci vývoje her.
<b>Cílová skupina</b>	Studenti středních škol na kurzu OOP4Fun. Základní znalosti programování včetně konceptů iterace a výběrů. Studenti by měli být seznámeni s programem Greenfoot.
<b>Trvání scénáře</b>	<ol style="list-style-type: none"> <li>1. Úloha 5.16 - Diskuse o tom, jak by měla instance třídy <b>Bullet</b> komunikovat s příslušnými objekty pomocí zpráv (15 min) .....</li> <li>2. Úloha 5.17 - Implementace zásahu instance třídy <b>Enemy</b> instancí třídy <b>Bullet</b> (30 min) .....</li> <li>3. Vysvětlení kódu metod <b>Greenfoot.showText</b>, <b>Greenfoot.getRandomNumber</b> a <b>World.act</b> (15 min) ..</li> <li>4. Úloha 5.18 - Vznik nepřátel a konec hry (30 min) .....</li> <li>5. Přehled o asociacích (20 min) .....</li> </ol>
<b>Materiály a zdroje</b>	<ul style="list-style-type: none"> <li>• učebnice z projektu OOP4Fun.</li> <li>• zdroje z projektu OOP4Fun.</li> </ul>

	<ul style="list-style-type: none"> <li>• zdrojový kód projektu z repozitáře Github/Gitlab.</li> <li>• Internetové zdroje.</li> </ul>
<p><i>Popis</i></p>	<p>V tomto výukovém scénáři, která trvá 110 minut, se studenti středních škol seznámí se specifickými principy dynamiky hry zahrnující střely a nepřátele v prostředí Greenfoot. Hodina se zaměřuje na rozvoj dovedností studentů při vytváření interaktivní a dynamické hry prostřednictvím efektivních interakcí objektů a herních mechanik.</p> <p>Hodina začíná 15 minutovou diskusí o tom, jakým způsobem by instance třídy <b>Bullet</b> měly reagovat na příslušné herní objekty pomocí zpráv. Tato diskuse představuje východisko pro pochopení toho, jakým způsobem objekty komunikují a spolupracují s cílem dosažení specifického chování hry.</p> <p>Následně se studenti budou 30 minut věnovat implementaci funkce, při které instance třídy <b>Bullet</b> úspěšně zasáhne instance třídy <b>Enemy</b>. Tato úloha je pro studenty výzvou, aby uplatnili své znalosti při detekci kolizí objektů a zpracování událostí na vytvoření působivých interakcí ve hře.</p> <p>Další segment obsahuje 15 minutové vysvětlení základních metod prostředí Greenfoot: <b>Greenfoot.showText(String, int, int)</b>, <b>Greenfoot.getRandomNumber(int)</b> a <b>World.act()</b>. Studenti získají přehled o tom, jakým způsobem tyto metody přispívají k zobrazování textu, generování náhodných čísel pro herní mechaniky a řízení cyklu aktualizace herního světa.</p> <p>Studenti potom stráví 30 minut řešením úloh týkajících se vytváření nepřátel a podmínek na ukončení hry. Patří sem návrh a implementace mechanismů na vytváření nepřátel ve vhodných intervalech a určení podmínek k ukončení hry na základě akcí hráče nebo cílů hry.</p> <p>Následuje 20 minutová kontrola, která se zaměřuje na upevnění pochopení asociací objektů a jejich úloh při realizaci herní dynamiky. Studenti si zopakují a zdokonalí své pochopení interakce a spolupráce objektů v prostředí Greenfoot s cílem dosáhnout požadované herní efekty.</p> <p>Na závěr lekce studenti hlouběji pochopí, jakým způsobem vytvářet interaktivní a poutavou herní dynamiku zahrnující střely, nepřátele a strategické herní mechaniky v prostředí Greenfoot. Budou vybaveni praktickými dovednostmi pro implementaci interakcí objektů, zprávu herních stavů a zlepšování zážitků hráčů prostřednictvím strukturovaných principů herního designu.</p>
<p><i>Hodnocení</i></p>	<p>Gamifikace představuje neformální nejen hodnocení, ale také zvýší zájem, vnitřní motivaci a vzdělávací výstupy celé skupiny.</p>
<p><i>Šíření výsledků</i></p>	<p>Na šíření výsledků mezi učitelem a studenty se použije repozitář Github/Gitlab a systém řízení výuky (např. Moodle). Studenti mohou pokračovat v diskusi na dané téma na fóru, které je jim poskytnuté prostřednictvím nástroje na řízení výuky.</p>

#### 7.4.1. Příručka učitele na přípravu vyučovací hodiny

1. Úloha 5.16 - Diskuse o tom, jak by měla instance třídy **Bullet** komunikovat s příslušnými objekty pomocí zpráv (15 min)

### Cíl:

Studenti jsou schopni kriticky a tvořivě diskutovat o tom, jak by měly střely reagovat na jiné objekty, zejména na nepřátele, prostřednictvím posílání zpráv.

### Pojmy na diskusi:

- posílání zpráv, detekce kolizí a interakce objektů

### Aktivity:

Na začátku se proberou interakce objektů v prostředí Greenfoot se zaměřením na to, jak instance různých tříd komunikují a jak se navzájem ovlivňují. Učitel zapojí studenty do diskuse s cílem posilnit tyto pojmy a jejich praktické využití při vývoji her.

Studenti diskutují o logice interakcí se střelami a o zprávách, které musí posílat. Učitel začne vysvětlováním důležitosti posílání zpráv v objektově orientovaném programování. Diskuse se potom zaměří na to, jak instance třídy **Bullet** reagují na jiné objekty, např. na nepřátele v aréně, zejména když střela zasáhne nepřátele.

Studenti jsou vyzváni, aby se v rámci diskuse podělili o své nápady týkající se logiky interakce se střelami. Uvažují o otázkách, jako například:

- Co by se mělo stát, když střela zasáhne nepřítele?
- Jak by měla střela oznámit tuto událost ostatním objektům?
- Jaké zprávy je potřebné odevzdat na správné zpracování interakce?

Učitel představí koncept detekce kolize a vysvětlí, jakým způsobem hra zjistí, kdy střela zasáhne nepřítele. Diskutuje se i o následných akcích, jako je snížení HP nepřítele nebo jeho odstranění z arény.

Na vizualizaci těchto interakcí učitel použije sekvenční diagram UML. Diagram znázorňuje zprávy, které se vymění mezi třídami **Bullet**, **Enemy** a **Arena** v průběhu interakce.

## 2. Úloha 5.17 - Implementace zásahu instance třídy **Enemy** instancí třídy **Bullet** (30 min)

### Cíl:

Učitel pomáhá studentům při realizaci interakce mezi střelou a nepřítelem.

### Pojmy na diskusi:

- detekce kolizí, volání metod a změny stavu objektu.

### Aktivity:

Studenti napíší kód na zpracování kolize mezi střelou a nepřítelem, včetně důsledků kolize. Učitel na začátku vysvětlí koncepty detekce kolize a volání metod v prostředí Greenfoot a zdůrazní, že pro zpracování interakcí mezi herními objekty jsou klíčové. Studenti budou krok za krokem vedeni k implementaci logiky kolize mezi střelami a nepříteli.

Studenti si nejprve připraví potřebné atributy a metody ve třídách **Bullet** a **Enemy**. Poté napíší kód ve třídě **Bullet** na detekci kolize s instancí třídy **Enemy** a zavolají metodu **hit(Bullet)** ze třídy **Enemy**. Dále implementují metodu **hit(Bullet)** ve třídě **Enemy** na zpracování účinků kolize, jako je snížení HP nepřítele nebo jeho odstranění ze hry. Studenti otestují svou implementaci, aby se ujistili, že interakce mezi střelou a nepřítelem funguje tak, jak má.

Na konci hodiny studenti budou mít funkční mechanismus detekce kolize mezi střelami a nepřáteli s příslušnými voláními metod a změnami stavu objektů. Toto cvičení posilí jejich pochopení detekce kolizí, volání metod a praktického využití těchto konceptů při vývoji her.

**Commit:** [dcfe31bc006b7f3dcd8b8b759cc1be901c32913c](https://github.com/4FUN/4FUN/commit/dcfe31bc006b7f3dcd8b8b759cc1be901c32913c)

### 3. Vysvětlení kódu metod `Greenfoot.showText`, `Greenfoot.getRandomNumber` a `World.act` (15 min)

#### Cíl:

Učitel vysvětlí použití konkrétních metod prostředí Greenfoot, které budou použité ve hře.

#### Pojmy na diskusi:

- zobrazení textu, generování náhodných čísel a metoda `act()`.

#### Aktivity:

Studenti se naučí zobrazovat text na obrazovce, generovat náhodná čísla a implementovat logiku hry v metodě `act()`.

Učitel začne představením metod `Greenfoot.showText(String, int, int)`, `Greenfoot.getRandomNumber(int)` a `World.act()`. Rozebere účel těchto metod a zdůrazní jejich význam při vývoji her na zobrazování informací, použití náhodnosti a definování chování objektů.

Metoda `Greenfoot.showText(String, int, int)` se používá na zobrazení textu na obrazovce na zadaných souřadnicích. Učitel vysvětlí, že tato metoda je užitečná na zobrazování informací o hře, jak je skóre, HP nebo pokynů přímo na obrazovce hry.

Metoda `Greenfoot.getRandomNumber(int)` vygeneruje náhodné číslo mezi 0 (včetně) a zadanou hodnotou, která představuje jeho horní hranici. Učitel diskutuje o tom, jak je možné tuto metodu použít na zavedení náhodnosti do hry, například na vytváření nepřátel na náhodných místech nebo na generování náhodných vzorů pohybu.

Metoda `World.act()` je opakovaně volána prostředím Greenfoot na vykonávání hlavní logiky hry. Učitel zdůrazňuje, že metoda `act()` je místem, kde jsou umístěné hlavní herní akce a logika hry, což umožňuje průběžné aktualizace a interakce ve hře.

### 4. Úloha 5.18 - Vznik nepřátel a konec hry (30 min)

#### Cíl:

Učitel pomáhá studentům implementovat vytváření nepřátel a podmínky na konci hry.

#### Pojmy na diskusi:

- tvorba objektů, herní smyčka a podmínky ukončení hry.

#### Aktivity:

Studenti napíší kód, který pravidelně vytváří nepřátele a definuje podmínky, které vyvolají konec hry. Učitel začne vysvětlováním důležitosti vytváření nepřátel v pravidelných intervalech a definováním podmínek pro ukončení hry, když jsou všichni nepřátelé poraženi. Rozeberou se pojmy vytváření objektů, herních smyček a podmínky pro ukončení hry. Díky tomu studenti jasně pochopí, co je potřebné implementovat.

Na opakované pravidelné volání procesu vytváření nepřátel se použije metoda **Arena.act()**. Učitel vysvětlí studentům, že by se měl zavést mechanismus zpoždění na kontrolu intervalu mezi vytvářením nepřátel. Studenti vytvoří metodu **spawn()** ve třídě **Arena**, která bude zpracovávat samotný proces vytváření nepřátel. Tato metoda vytvoří instanci třídy **Enemy**, přiřadí jí vlastnosti (např. pozici) a přidá ji do arény.

Studenti definují a implementují podmínky na ukončení hry. Když počet nepřátel klesne na nulu, hra se ukončí a hráč vyhraje. Na tento účel by studenti měli ve třídě **Arena** udržovat atribut, který bude sledovat počet vytvořených nepřátel. Tento atribut se zvýší při každém vytvoření nepřítele a sníží, pokud je nepřítel zabítý. Metoda **Arena.kill(Enemy)** by měla být upravená tak, aby kontrolovala, zda jsou všichni nepřátelé zabítý. Pokud k této situaci dojde, studenti zastaví hru pomocí metody **Greenfoot.stop()** a zobrazí zprávu o vítězství. Volání **Greenfoot.stop()** by proto mělo být posledním příkazem v metodě.

Na konci studenti otestují mechanismus vytváření nepřátel pozorováním jejich pravidelného vytváření v aréně. Ujistí se, že zpoždění mezi jejich vytvořeními funguje správně. Ověří stav konce hry simulací porážky všech nepřátel a zkontroluje, zda se hra zastaví se zobrazenou zprávou o výhře.

Na konci této lekce budou studenti mít implementovaný funkční systém vytváření nepřátel a definované jasné podmínky pro ukončení hry, čímž si upevní své znalosti o herních smyčkách, správě objektů a výsledcích hry založených na podmínkách.

**Commit:** [d48341a095561500af6032d5c8f56e201060f9a4](https://github.com/4FUN/4FUN/commit/d48341a095561500af6032d5c8f56e201060f9a4)

## 5. Přehled o asociacích (20 min)

### Cíl:

Učitel zopakuje koncept asociací mezi třídami a zdůrazní, jakým způsobem na sebe reagují objekty v prostředí Greenfoot a jakým způsobem komunikují.

### Pojmy na diskusi:

- asociace, komunikace objektů a posílání zpráv.

### Aktivita:

Studenti diskutují a ověřují si, jak pochopili asociaci, přičemž si zakreslují souvislosti mezi různými objekty a jejich vzájemným působením.

Učitel začne zopakováním klíčových pojmů týkajících se asociací mezi třídami. Patří sem i to, jak na sebe objekty vzájemně působí, jak vzájemně komunikují a jak si posílají zprávy. Na ulehčení tohoto postupu může učitel vycházet z příkladů a úloh probraných dříve, čímž studentům pomůže upevnit si vědomosti a pochopit, jakým způsobem se tyto pojmy uplatňují v prostředí Greenfoot.

Na vizuální znázornění interakcí mezi různými objekty použije učitel sekvenční diagramy UML. Ukáže například posloupnost zpráv, když střela zasáhne nepřítele, a jakým způsobem aréna zpracovává vytváření a odstraňování nepřátel.

Na konci této lekce budou studenti mít upevněné vědomosti o asociacích a interakcích objektů v prostředí Greenfoot. Budou schopni jasně vyjádřit, jak různé objekty v hře komunikují a spolupracují, a aplikovat tyto koncepty na svoje vlastní herní projekty.

## 8. Dědičnost

V rámci tematického celku dědičnost byly vytvořené čtyři metodické materiály.

## 8.1. Úvod do dědičnosti v prostředí Greenfoot

Tabulka 18. Úvod do dědičnosti v prostředí Greenfoot

<i>Název</i>	Úvod do dědičnosti v prostředí Greenfoot
<i>Cíle vzdělávání</i>	Na konci této kapitoly budou studenti schopni porozumět konceptům dědičnosti. Pochopení zkoumaných pojmů se bude rozebírat v kontextu vývoje her, což podpoří kreativitu, týmovou práci a nadšený přístup k programování s využitím nástroje Greenfoot.
<i>Cílová skupina</i>	Studenti středních škol na kurzu OOP4Fun. Základné znalosti programování a základní znalosti objektově orientovaného programování. Studenti by měli být seznámeni s programem Greenfoot.
<i>Trvání scénáře</i>	<ol style="list-style-type: none"> <li>1. Základní pojmy dědičnosti (15 min).....</li> <li>2. Hierarchie tříd a dědičnost (15 min).....</li> <li>3. Úloha 6.1 a 6.2: Identifikace společných vlastností a identifikace třídy předka (30 min).....</li> <li>4. Úvod do abstraktních tříd (5 min).....</li> <li>5. Úloha 6.3: Definice abstraktní třídy ve hře (10 min).....</li> </ol>
<i>Materiály a zdroje</i>	<ul style="list-style-type: none"> <li>• učebnice z projektu OOP4Fun</li> <li>• zdroje z projektu OOP4Fun.</li> <li>• zdrojový kód projektu z repozitáře Github/Gitlab.</li> <li>• internetové zdroje.</li> </ul>
<i>Popis</i>	<p>V tomto 75-minutovém výukovém scénáři se studenti středních škol seznámí s principy objektově orientovaného programování, které souvisí s dědičností z pohledu vývoje her pomocí nástroje Greenfoot. Vyučování začne 15-minutovým úvodem učitele o základních konceptech dědičnosti, přičemž se vytvoří kontext související s předchozími kapitolami a budoucím vývojem hry.</p> <p>Potom následuje 15-minutový úsek, během kterého studenti a učitel diskutují o hierarchii tříd v jejich hře. S cílem vysvětlit pojmy související s dědičností se pozorují třídy <b>Orb</b> a <b>Direction</b>. Během dalších 15-minut se zkoumá identifikace společných vlastností těchto tříd. Pozoruje se, že instance těchto tříd během své životnosti života nic nevykonávají, ale jen reagují na zprávy. Následně se identifikuje společná metoda pro vykonávání - metoda <b>act()</b>. Tato metoda bude v obou třídách definovaná s prázdným tělem. Na základě identifikovaných společných vlastností bude v následujících 15 minutách implementovaná nová třída <b>PassiveActor</b>, která bude obsahovat zatím prázdnou metodu <b>act()</b>.</p> <p>V dalším 5-minutovém úseku se představí abstraktní třídy. Abstraktní třídy slouží jako vzory pro jiné třídy a z toho důvodu není možné tvořit jejich instance. Jsou však nevyhnutelné při navrhování hierarchií tříd. Vzhledem k tomu, že třída <b>PassiveActor</b> je vzorem, tak v následujícím 10-minutovém úseku je definovaná jako abstraktní třída. Navíc je třída <b>PassiveActor</b> ustanovená jako předek tříd <b>Orb</b> a <b>Direction</b>, čímž se třídy <b>Orb</b> a <b>Direction</b> stávají jejími potomky. Jelikož metoda <b>act()</b> je už definovaná ve třídě <b>PassiveActor</b>, je z tříd <b>Orb</b> a <b>Direction</b> odstraněná.</p> <p>Výsledkem je, že na konci hodiny se studenti seznámí s novými pojmy souvisejícími s</p>

	dědičností.
<i>Hodnocení</i>	<p>Gamifikace představuje nejen neformální hodnocení, ale také zvýší zájem, vnitřní motivaci a výsledky vzdělávání celé skupiny.</p> <p>Vzhledem na důležitost konceptu dědičnosti otevírá struktura projektu možnosti pro další diskusi a modifikaci. V této souvislosti je možné uvažovat o dalších třídách a jejich hierarchii a zavést další třídy, metody a atributy. Na druhé straně může učitel toto téma upravit tak, aby ukázal výhody dědičnosti a s ní spojenou univerzálnost jen na hierarchiích navržených v této knize.</p>
<i>Šíření výsledků</i>	Na šíření výsledků mezi učitelem a studenty se použije repozitář Github/Gitlab a systém řízení výuky (např. Moodle). Studenti mohou pokračovat v diskusi na dané téma na fóru, které je jim poskytnuté prostřednictvím nástroje na řízení výuky.

### 8.1.1. Příručka učitele na přípravu vyučovací hodiny

#### 1. Základní pojmy dědičnosti (15 min)

##### **Cíl:**

Studenti se seznámí s konceptem dědičnosti na základě příkladů z reálného života a diskuse o jeho výhodách.

##### **Pojmy na diskusi:**

- dědičnost, příklady dědičnosti v reálném světě.

##### **Aktivita:**

V úvodní části učitel uvede kontext, který souvisí s předchozími kapitolami. Dále učitel představí koncept dědičnosti. Měl by jej studentům přiblížit pomocí příkladů z reálného života. Např. pokud se uvažuje o vztahu rodič - dítě, tak děti dědí vlastnosti po svých rodičích, jako je typ vlasů, barva očí atd. Učitel by se studenty měl diskutovat o výhodách dědičnosti. Tyto pojmy se budou posuzovat v kontextu prostředí Greenfoot a programovacího jazyka Java.

#### 2. Hierarchie tříd a dědičnost (15 min)

##### **Cíl:**

Představení hierarchie tříd v kontextu dědičnosti prostřednictvím vysvětlení tříd předků a potomků, diskuse o reálných příkladech a vlastností dědičnosti, zdůraznění výhod hierarchie tříd.

##### **Pojmy na diskusi:**

- dědičnost, hierarchie tříd, příklady hierarchie tříd v reálném světě, výhody dědičnosti a hierarchie tříd.



**Aktivita:**

Učitel představí hierarchie tříd v kontextu konceptu dědičnosti. Učitel představí třídy předků (známe též jako: nadtřídy, rodičovské třídy) a třídy potomků (známé též jako: podtřídy, dětské třídy):

- V tomto kontextu je možné diskutovat o dříve prozkoumaných reálných třídách.
- V tomto kontextu by se mělo diskutovat o tom, že podtřídy mohou dědit vlastnosti (tj. atributy a metody) od rodičovské třídy.
- Dále by se mělo diskutovat o tom, že podtřídy mohou obsahovat další vlastnosti tj. ty, které nejsou dostupné v rodičovské třídě.
- Měli by se prodiskutovat výhody hierarchie tříd v kontextu koncepce dědičnosti. Mělo by se vysvětlit, že v programovacím jazyce Java může mít každá třída víc podtříd, ale len jednu rodičovskou třídu.

### 3. Úloha 6.1 a 6.2: Identifikace společných vlastností a identifikace třídy předka (30 min)

**Cíl:**

Identifikace společných vlastností ve třídách hry, nalezení třídy předka a implementace nové třídy v hierarchii tříd.

**Pojmy na diskusi:**

- dědičnost, hierarchie tříd, implementace dědičnosti a hierarchie tříd při vývoji her.

**Aktivita:**

V kontextu vývoje her se uvažuje o třídách **Orb** a **Direction**. Je potřebné poznamenat, že tyto třídy reagují na zprávy. Proto by se měla identifikovat společná metoda pro vykonávání, metoda **act()**. Na základě identifikovaných společných vlastností by se měla implementovat nová třída **PassiveActor** obsahující metodu **act()**:

- Tyto třídy (**PassiveActor**, **Orb** a **Direction**) by se měly použít na prezentaci hierarchie tříd v kontextu dědičnosti.
- Učitel může vizuálně znázornit hierarchii tříd pomocí diagramu hierarchie.
- Učitel upozorní studenty, co se změnilo v prostředí Greenfoot, tedy že je v kódu (hlavičce třídy) třída **Actor** nahrazena třídou **PassiveActor**

**Commit:** [afe617814c07a5d885ed06479bf71deda8725f19](https://github.com/afe617814c07a5d885ed06479bf71deda8725f19)

### 4. Úvod do abstraktních tříd (5 min)

**Cíl:**

Představení konceptu abstraktních tříd, diskuse o jejich úloze jako vzorů při navrhování hierarchií tříd a prozkoumání reálných příkladů na ilustraci jejich použití

**Pojmy na diskusi:**

- dědičnost, hierarchie tříd, abstraktní třídy, reálné příklady abstraktních tříd v kontextu dědičnosti a hierarchie tříd.

**Aktivita:**

Učitel zavede pojem abstraktní třída. Je potřebné diskutovat o tom, že abstraktní třídy slouží hlavně jen jako vzory pro jiné třídy a není možné tvořit jejich instance. Jsou však nevyhnutné při navrhování hierarchií tříd.

Učitel a studenti mohou diskutovat o příkladech z reálného světa, které se týkají abstraktních tříd a podtříd (např. třída počítač se základními vlastnostmi může být definovaná jako abstraktní třída a může být specializovaná na konzolu, stolní počítač, přenosný počítač a mobilní telefon, přičemž každá z nich má specifickou sadu vlastností atd.) Dalším příkladem mohou být geometrické útvary. Obdélník a trojúhelník je možné definovat jako potomka z abstraktní třídy geometrický útvar. Při výpočtu obvodu a obsahu obecného geometrického tvaru nemáme k dispozici přesný vzorec. Ale pro obdélník a trojúhelník přesný vzorec máme. Čtverec je možné definovat jako potomek obdélníka. Studenti by měli diskutovat o dalších příkladech geometrických útvarů a těles.

#### 5. Úloha 6.3: Definice abstraktní třídy ve hře (10 min)

##### Cíl:

Diskuse o úloze třídy **PassiveActor** a implementaci této třídy jako abstraktní.

##### Pojmy na diskusi:

- dědičnost, hierarchie tříd, abstraktní třídy, implementace abstraktní třídy při vývoji hry.

##### Aktivita:

Koncept abstraktní třídy je zohledněn v prostředí Greenfoot a programovacím jazyce Java. V kontextu vývoje her je třída **PassiveActor** vzorem pro hráče. Proto je definovaná jako abstraktní třída a ustanovená jako předek tříd **Orb** a **Direction**, čímž se třídy **Orb** a **Direction** stávají jejími potomky. Protože metoda **act()** je už definovaná v třídě **PassiveActor**, měla by se z tříd **Orb** a **Direction** odstranit.

**Commit:** [f7a5702cae29bf21c9c88620d01ef64e4127c21c](https://github.com/4FUN/4FUN/commit/f7a5702cae29bf21c9c88620d01ef64e4127c21c)

## 8.2. Koncepty dědičnosti v prostředí Greenfoot (část 1)

Tabulka 19. Koncepty dědičnosti v prostředí Greenfoot (část 1)

<b>Název</b>	Koncepty dědičnosti v prostředí Greenfoot (část 1)
<b>Cíle vzdělávání</b>	Na konci této kapitoly budou studenti rozumět dalším konceptům dědičnosti. Zkoumané pojmy se budou rozebírat v kontextu vývoje her, čímž se podpoří tvořivé myšlení, týmová práce a nadšený přístup k programování v prostředí Greenfoot.
<b>Cílová skupina</b>	Studenti středních škol, kteří se účastní kurzu OOP4Fun. Základní znalosti programování a základní znalosti objektově orientovaného programování. Studenti by měli být seznámeni s programem Greenfoot.
<b>Trvání scénáře</b>	<ol style="list-style-type: none"> <li>1. Úloha 6.4.: Identifikace společných vlastností souvisejících s pohybem entit (15 min) .....</li> <li>2. Úloha 6.5.: Definice abstraktní třídy týkající se pohybu entit (15 min) .....</li> <li>3. Úloha 6.6.: Identifikace vlastností specifických pro třídu, která souvisí s pohybem entit (15 min).....</li> <li>4. Úvod do klíčového slova super v kontextu dědičnosti (20 min) .....</li> <li>5. Úloha 6.7.: Refaktorování kódu souvisejícího s pohybem entit (30 min) .....</li> </ol>
<b>Materiály a zdroje</b>	<ul style="list-style-type: none"> <li>• učebnice z projektu OOP4Fun</li> <li>• zdroje z projektu OOP4Fun.</li> <li>• zdrojový kód projektu z repozitáře Github/Gitlab.</li> </ul>

	<ul style="list-style-type: none"> <li>internetové zdroje.</li> </ul>
<i>Popis</i>	<p>V průběhu tohoto 95-minutového výukového scénáře se studenti středních škol seznámí s pokročilými koncepcemi souvisejícími s dědičností pomocí nástroje Greenfoot.</p> <p>Hodina začne 15-minutovým úsekem, ve kterém se budou zkoumat běžné vlastnosti související s pohybem entit se zaměřením na třídy <b>Bullet</b> a <b>Enemy</b>. Tyto třídy v době svého životního cyklu chovají podobně. Pohybují se stejným způsobem a stejně reagují na okolí.</p> <p>Na základě zjištěných společných vlastností se bude v další 15-minutové úloze implementovat nová abstraktní třída <b>MovingActor</b> obsahující metodu <b>act()</b>. Tato třída je společným předkem tříd <b>Bullet</b> a <b>Enemy</b> a bude implementovat metodu <b>act()</b>, aby se podtřídy pohybovaly stejným způsobem a aby se zaměřily na svůj specifický účel.</p> <p>V dalším 15-minutové části se zkoumají vlastnosti specifické pro třídy související s pohybem entit. V této souvislosti se zkoumá metoda <b>act()</b> příslušných tříd. Zejména její atributy <b>moveDelay</b> a <b>nextMoveCounter</b>. Na závěr této části je možné konstatovat, že kód metody <b>act()</b> zodpovědný za pohyb je stejný.</p> <p>Potom následuje 20-minutový blok, v průběhu kterého bude představené klíčové slovo <b>super</b> v kontextu dědičnosti.</p> <p>V posledním 30-minutovém úseku se provede refaktorizace kódu související pohybem entit. Výsledkem je, že dříve identifikované atributy <b>moveDelay</b> a <b>nextMoveCounter</b> se přesunou z podtříd <b>Bullet</b> a <b>Enemy</b> jejich společného předka třídy <b>MovingActor</b>. Navíc je ve třídě <b>MovingActor</b> definovaný parametrický konstruktor na inicializaci těchto atributů. Tento konstruktor s příslušnými parametry je z podtříd <b>Bullet</b> a <b>Enemy</b> vyvolán pomocí klíčového slova <b>super</b>. Stejně tak je kód odpovědný za pohyb v metodě <b>act()</b> podtříd <b>Bullet</b> a <b>Enemy</b> přesunutý do metody <b>act()</b> třídy <b>MovingActor</b>, přičemž zbytek implementace zůstal v podtřídách nezměněný. Nakonec se rodičovská verze metody <b>act()</b> volá jako první řádek metody <b>act()</b> v podtřídách <b>Bullet</b> a <b>Enemy</b> pomocí klíčového slova <b>super</b>.</p> <p>Výsledkem výukového scénáře je, že se studenti seznámí s novými pojmy souvisejícími s dědičností.</p>
<i>Hodnocení</i>	<p>Gamifikace představuje nejen neformální hodnocení, ale také zvýší zájem, vnitřní motivaci a výsledky vzdělávání celé skupiny.</p> <p>Vzhledem na důležitost konceptu dědičnosti otevírá struktura projektu možnosti pro další diskusi a modifikaci. V této souvislosti je možné uvažovat o dalších třídách a jejich hierarchii a zavést další třídy, metody a atributy. Na druhé straně může učitel toto téma upravit tak, aby ukázal výhody dědičnosti a s ní spojenou univerzálnost jen na hierarchiích navržených v této knize.</p>
<i>Šíření výsledků</i>	<p>Na šíření výsledků mezi učitelem a studenty se použije repozitář Github/Gitlab a systém řízení výuky (např. Moodle). Studenti mohou pokračovat v diskusi na dané téma na fóru, které je jim poskytnuté prostřednictvím nástroje na řízení výuky.</p>

### 8.2.1. Příručka učitele na přípravu vyučovací hodiny

#### 1. Úloha 6.4.: Identifikace společných vlastností souvisejících s pohybem entit (15 min)

**Cíl:**

Studenti zkoumají třídy **Bullet** a **Enemy** s důrazem na jejich podobné chování v době své existence ve světě, zejména na to, jak se pohybují a jak reagují na okolí.

**Pojmy na diskusi:**

- dědičnost, hierarchie tříd, abstraktní třídy.

**Aktivita:**

Učitel a studenti se zaměří na třídy **Bullet** a **Enemy**, které se v době své existence ve světě chovají podobně. Je třeba si všimnout, že tyto se třídy pohybují stejným způsobem a stejným způsobem reagují na okolí.

## 2. Úloha 6.5.: Definice abstraktní třídy týkající se pohybu entit (15 min)

**Cíl:**

Zkoumání tříd **Bullet** a **Enemy** s důrazem na jejich podobné chování v době své existence ve světě, zejména zkoumání jejich pohybu a reakcí na okolí.

**Pojmy na diskusi:**

- dědičnost, hierarchie tříd, abstraktní třídy, implementace abstraktní třídy při vývoji hry.

**Aktivita:**

Na základě identifikovaných společných vlastností by měla být implementována nová abstraktní třída **MovingActor** obsahující metodu **act()**. Tato třída se vytvoří jako společný předek tříd **Bullet** a **Enemy**, čímž se třídy **Bullet** a **Enemy** stanou jejími potomky. Mělo by se diskutovat o tom, že podtřídy dědí společné vlastnosti z rodičovské třídy. Tím pádem je třída **MovingActor** vzorem pro návrh třídy a měla by se deklarovat jako abstraktní.

**Commit:** [43e53b533563ce0a860b294ad9009f77409c48d4](https://github.com/4FUN/4FUN/commit/43e53b533563ce0a860b294ad9009f77409c48d4)

## 3. Úloha 6.6.: Identifikace vlastností specifických pro třídu, která souvisí s pohybem entit (15 min)

Zkoumání tříd **Bullet** a **Enemy** s důrazem na jejich podobné chování v době své existence, zejména zkoumání jejich pohybu a reakcí na okolí.

**Pojmy na diskusi:**

- dědičnost, hierarchie tříd, abstraktní třídy.

**Aktivita:**

Učitel a studenti zkoumají vlastnosti specifické pro třídu, které souvisí s pohybem entit. Hlavně se zkoumá metoda **act()** příslušných tříd a také atributy **moveDelay** a **nextMoveCounter**. Ze zkoumání by mělo vyplynout, že zdrojový kód metody **act()** odpovědný za pohyb je u různých entit stejný.

#### 4. Úvod do klíčového slova **super** v kontextu dědičnosti (20 min)

##### Cíl:

Představení klíčového slova **super** v kontextu dědičnosti, demonstrace jeho využití na volání vlastností z rodičovské třídy a diskuse o jeho výhodách.

##### Pojmy na diskusi:

- dědičnost, hierarchie tříd, klíčové slovo **super** v kontextu dědičnosti. Pozice příkazu **super**.

##### Aktivita:

Učitel představí klíčové slovo **super** v kontextu dědičnosti. Dále vysvětlí možnosti jeho využití:

- klíčové slovo **super** je možné použít na vyvolání konstrukturu z nadřazené třídy,
- klíčové slovo **super** je možné použít vyvolání metody z nadřazené třídy,
- klíčové slovo **super** je možné použít vyvolání atributu z nadřazené třídy,
- klíčové slovo **super** musí být použité jako první příkaz v konstrukturu.

Je potřebné diskutovat o výhodách používání klíčového slova **super** v kontextu dědičnosti.

#### 5. Úloha 6.7.: Refaktorování kódu souvisejícího s pohybem entit (30 min)

##### Cíl:

Refaktorování kódu souvisejícího s pohybem entit z podtříd **Bullet** a **Enemy** do rodičovské třídy **MovingActor**.

##### Pojmy na diskusi:

- dědičnost, hierarchie tříd, klíčové slovo **super** v kontextu dědičnosti.

##### Aktivita:

Učitel se studenty provede refaktORIZACI kódu, který se týká pohybu entit. Atributy **moveDelay** a **nextMoveCounter** se přesunou z podtříd **Bullet** a **Enemy** do třídy jejich předka - třídy **MovingActor**. Parametrický konstrukturu na inicializaci těchto atributů je nově definovaný ve třídě **MovingActor**. Tento konstrukturu s příslušnými parametry je nově vyvolán z podtříd **Bullet** a **Enemy** pomocí klíčového slova **super**. Kód odpovědný za pohyb v metodě **act()** je přesunutý z podtříd **Bullet** a **Enemy** do metody **act()** třídy **MovingActor**. Zbytek implementace zůstal v podtřídách **Bullet** a **Enemy** nezměněný. Nakonec se rodičovská verze metody **act()** volá jako pomocí klíčového slova **super** v prvním řádku metody **act()** v podtřídách **Bullet** a **Enemy**. Na závěr je třeba diskutovat o tom, že podtřídy mohou obsahovat další vlastnosti, které nejsou dostupné v rodičovské třídě (tj. jiná implementace metody **act()**).

Commit: [ca1f010a63445c1847b74259a1c6cd4817121db3](https://github.com/4FUN/4FUN/commit/ca1f010a63445c1847b74259a1c6cd4817121db3)

### 8.3. Koncepty dědičnosti v prostředí Greenfoot (část 2)

Tabulka 20. Koncepty dědičnosti v prostředí Greenfoot (část 2)

Název	Koncepty dědičnosti v prostředí Greenfoot (část 2)
Cíle vzdělávání	Na konci této kapitoly budou studenti rozumět dalším konceptům dědičnosti. Prozkoumávané pojmy se budou rozebírat v kontextu vývoje her, čímž se podpoří tvořivé

	myšlení, týmová práce a nadšený přístup ke kódování v prostředí Greenfoot.
<i>Cílová skupina</i>	Studenti středních škol, kteří se účastní kurzu OOP4Fun. Základní znalosti programování a základní znalosti objektově orientovaného programování. Studenti by měli být seznámeni s programem Greenfoot.
<i>Trvání scénáře</i>	<ol style="list-style-type: none"> <li>1. Úloha 6.8.: Vytváření vlastních nepřátel (30 min) .....</li> <li>2. Liskové principy zastoupení (20 min).....</li> <li>3. Úloha 6.9.: Vytváření vlastních nepřátel (20 min) .....</li> </ol>
<i>Materiály a zdroje</i>	<ul style="list-style-type: none"> <li>• učebnice z projektu OOP4Fun</li> <li>• zdroje z projektu OOP4Fun.</li> <li>• zdrojový kód projektu z repozitáře Github/Gitlab.</li> <li>• internetové zdroje.</li> </ul>
<i>Popis</i>	<p>V průběhu tohoto 70-minutového výukového scénáře se studenti středních škol seznámí s pokročilými koncepcemi dědičnosti s využitím nástroje Greenfoot.</p> <p>Výuka začne 30-minutovým úsekem zaměřeným na třídu <b>Enemy</b>, kde studenti vytvářejí další podtřídy, které představují různé nepřátele - např. třídy <b>Frog</b> (žába) a <b>Spider</b> (Pavouk). V této souvislosti jsou pro každý typ nepřítele definované obrázky a bezparametrické konstruktory (s příslušným voláním rodičovského konstruktora).</p> <p>V dalším 20-minutovém bloku se představí Liskové princip zastoupení (angl. Liskov Substitution Principle, dále LSP). Tento princip, který je součástí principů SOLID objektově orientovaného návrhu uvádí, že funkce, které používají ukazovatele nebo odkazy na nadřazené třídy, by měly být schopné používat objekty podtříd.</p> <p>Následuje poslední 20-minutová úloha, která je věnovaná vytváření vlastních nepřátel. Zkoumá se metoda <b>Arena.spawn()</b> a prostřednictvím různých rozhodnutí se vytvářejí vlastní nepřátele a ukládají se do proměnné typu <b>Enemy</b>. Pozoruje se, že v aplikaci není nutné měnit žádný jiný kód, což dokazuje aplikaci LSP.</p> <p>Výsledkem je, že na konci hodiny se studenti seznámí s pokročilými koncepty dědičnosti a praktickými aplikacemi.</p>
<i>Hodnocení</i>	<p>Gamifikace představuje nejen neformální hodnocení, ale také zvýší zájem, vnitřní motivaci a výsledky vzdělávání celé skupiny.</p> <p>Vzhledem na důležitost konceptu dědičnosti otevírá struktura projektu možnosti pro další diskusi a modifikaci. V této souvislosti je možné uvažovat o dalších třídách a jejich hierarchii a zavést další třídy, metody a atributy. Na druhé straně může učitel toto téma upravit tak, aby ukázal výhody dědičnosti a s ní spojenou univerzálnost jen na hierarchiích navržených v této knize.</p>
<i>Šíření výsledků</i>	Na šíření výsledků mezi učitelem a studenty se použije repozitář Github/Gitlab a systém řízení výuky (např. Moodle). Studenti mohou pokračovat v diskusi na dané téma na fóru, které je jim poskytnuté prostřednictvím nástroje na řízení výuky.

### 8.3.1. Příručka učitele na přípravu vyučovací hodiny

#### 1. Úloha 6.8.: Vytváření vlastních nepřátel (30 min)

**Cíl:**

Studenti definují podtřídy třídy **Enemy** s vhodnými s obrázky a bezparametrickými konstruktory, které vhodně vyvolávají rodičovský konstruktorem.

**Pojmy na diskusi:**

- dědičnost, hierarchie tříd, klíčové slovo **super** v kontextu dědičnosti.

**Aktivita:**

Studenti se zaměří na třídu **Enemy** a definici dalších podtříd reprezentujících různé nepřátele (např. **Frog** a **Spider**). Pro každý typ nepřátel by měly být definované obrázky a bezparametrické konstruktory (s příslušným voláním rodičovského konstruktora).

**Commit:** [b0ac1fbe793548a32f7700c292aed631918c8388](https://github.com/8P/4FUN/commit/b0ac1fbe793548a32f7700c292aed631918c8388)

#### 2. Liskové principy zastoupení (20 min)

**Cíl:**

Představení LSP, diskuse o reálném světě a zkoumání výhod dodržování tohoto principu v kontextu dědičnosti.

**Pojmy na diskusi:**

- dědičnost, hierarchie tříd, referenční proměnné, LSP (Liskové princip zastoupení).

**Aktivita:**

Zavádí se LSP. Tento princip je součástí SOLID principů objektově orientovaného návrhu. Princip uvádí, že funkce, které používají ukazatele nebo odkazy na nadřazené třídy, by měli být schopné používat i objekty podtříd. Měli by se probrat reálné příklady (např. že pokud je třída **Computer** definovaná jako rodičovská třída a třídy **Console**, **Desktop**, **Laptop** a **Mobile Phone** jsou definované jako podtřídy, LSP uvádí, že funkce, které používají třídu **Computer**, budou umět pracovat se všemi podtřídami bez jakékoliv změny v jejich kódu). Měly by se prodiskutovat výhody použití LSP v kontextu dědičnosti.

#### 3. Úloha 6.9.: Vyvolávání vlastních nepřátel (20 min)

**Cíl:**

Demonstrace aplikace LSP vytvořením vlastních nepřátel.

**Pojmy na diskusi:**

- dědičnost, hierarchie tříd, referenční proměnné, LSP.

**Aktivita:**

Tato úloha je určená pro vytváření vlastních nepřátel. Zkoumá se metoda **Arena.spawn()** a prostřednictvím různých rozhodnutí se vytvářejí vlastní nepřátele a ukládají se do proměnné typu **Enemy**. Učitel upozorní studenty, že je potřeba všimnout si, že v aplikaci není nutné měnit žádný jiný kód, což dokazuje uplatnění LSP.

**Commit:** [8cd4397f585ec957bbc18ca98e01823f434a13a6](https://github.com/8P/4FUN/commit/8cd4397f585ec957bbc18ca98e01823f434a13a6)

## 8.4. Koncepty dědičnosti v prostředí Greenfoot (část 3)

Tabulka 21. Koncepty dědičnosti v prostředí Greenfoot (část 3)

<b>Název</b>	Koncepty dědičnosti v prostředí Greenfoot (část 3)
<b>Cíle vzdělávání</b>	Na konci této části budou studenti rozumět dalším konceptům dědičnosti. Prozkoumané pojmy se budou rozebírat v kontextu vývoje her, čímž se podpoří tvořivé myšlení, týmová práce a nadšený přístup ke kódování v prostředí Greenfoot.
<b>Cílová skupina</b>	Studenti středních škol, kteří se účastní kurzu OOP4Fun. Základní znalosti programování a základní znalosti objektově orientovaného programování. Studenti by měli být seznámeni s programem Greenfoot.
<b>Trvání scénáře</b>	<ol style="list-style-type: none"> <li>1. Úloha 6.10.: Diskuse o hierarchii arén (20 min).....</li> <li>2. Úloha 6.11. a 6.12.: Vytvořte univerzální arénu a vytvořte DemoArenu (45 min) .....</li> <li>3. Úloha 6.13.: Vytvoření vlastních arén (30 min) .....</li> <li>4. Opakování teorie dědičnosti (20 min) .....</li> </ol>
<b>Materiály a zdroje</b>	<ul style="list-style-type: none"> <li>• učebnice z projektu OOP4Fun</li> <li>• zdroje z projektu OOP4Fun.</li> <li>• zdrojový kód projektu z repozitáře Github/Gitlab.</li> <li>• internetové zdroje.</li> </ul>
<b>Popis</b>	<p>V průběhu tohoto 115-minutového výukového scénáře se studenti středních škol seznámí s pokročilými koncepcemi dědičnosti pomocí nástroje Greenfoot.</p> <p>Začne se 20-minutovou diskusí o hierarchii arény. Podtřídy třídy <b>Arena</b> jsou odpovědné za vlastní rozložení (např. pozice instancí <b>Orb</b> a <b>Direction</b>, velikost arény). Tyto úlohy se vykonávají v konstruktorech podtříd, které nastavují a ukládají pozice místa vzniku, rotaci instancí a rozměry arény.</p> <p>V následující 30-minutové úloze je zavedená univerzální třída <b>Arena</b>. Definují se další atributy (<b>spawnPositionX</b>, <b>spawnPositionY</b> a <b>spawnRotation</b>), které se inicializují v konstruktoru a používají se v metodách <b>spawn()</b> a <b>respawn(Enemy)</b>. Atributy týkající se rozměrů arény (šířka a výška) jsou též definované a inicializované v konstruktoru. Jelikož třída <b>Arena</b> slouží jako vzor na definování konkrétních arén, je definována jako abstraktní třída.</p> <p>Na základě identifikované třídy <b>Arena</b> se v další 15-minutové úloze představí podtřída <b>DemoArena</b>. Definuje se konstruktor <b>DemoArena</b>, který volá konstruktor nadřazené třídy, a kód odpovědný za rozložení instancí tříd <b>Orb</b>, <b>Tower</b> a <b>Direction</b> se přesune z konstruktoru třídy <b>Arena</b> do konstruktoru třídy <b>DemoArena</b>. Nakonec se vytvoří nová instance třídy <b>DemoArena</b>.</p> <p>V následujícím 30-minutovém úseku se vytvoří další inovativní podtřídy třídy <b>Arena</b>. Kód je možné sdílet s ostatními studenty ve skupině.</p> <p>Poslední 20-minutová fáze zabývá teorií, která souvisí s dědičností.</p> <p>Výsledkem je, že na konci hodiny se studenti seznámí s pokročilými koncepty dědičnosti a</p>



	jejich praktickými aplikacemi.
<i>Hodnocení</i>	<p>Gamifikace představuje nejen neformální hodnocení, ale také zvýší zájem, vnitřní motivaci a výsledky vzdělávání celé skupiny.</p> <p>Vzhledem na důležitost konceptu dědičnosti otevírá struktura projektu možnosti pro další diskusi a modifikaci. V této souvislosti je možné uvažovat o dalších třídách a jejich hierarchii a zavést další třídy, metody a atributy. Na druhé straně může učitel toto téma upravit tak, aby ukázal výhody dědičnosti a s ní spojenou univerzálnost jen na hierarchiích navržených v této knize.</p>
<i>Šíření výsledků</i>	Na šíření výsledků mezi učitelem a studenty se použije repozitář Github/Gitlab a systém řízení výuky (např. Moodle). Studenti mohou pokračovat v diskusi na dané téma na fóru, které je jim poskytnuté prostřednictvím nástroje na řízení výuky.

#### 8.4.1. Příručka učitele na přípravu vyučovací hodiny

##### 1. Úloha 6.10.: Diskuse o hierarchii arén (20 min)

###### Cíl:

Studenti prozkoumají hierarchie tříd **Arena** s důrazem na to, jak jsou podtřídy odpovědné za definování vlastních rozložení a implementaci těchto rozložení v rámci jejich příslušných konstruktorů.

###### Pojmy na diskusi:

- dědičnost, hierarchie tříd, konstruktory.

###### Aktivita:

Hierarchie tříd arény je prodiskutovaná. Je nutné poznamenat, že podtřídy arény jsou odpovědné za její vlastní rozložení (např. pozice instancí tříd **Orb**, **Direction** a velikost arény). Tyto úlohy se vykonávají v konstruktorech podtříd, které nastavují a ukládají místa vzniku instancí, jejich rotace a rozměry arény.

##### 2. Úloha 6.11. a 6.12.: Vytvořte univerzální arénu a vytvořte DemoArenu (45 min)

###### Cíl:

Představení univerzální abstraktní třídy **Arena**, definování a inicializace konkrétní podtřídy třídy **Arena**, prozkoumání hierarchie tříd **Arena**.

###### Pojmy na diskusi:

- dědičnost, hierarchie tříd, abstraktní třídy, konstruktory.

###### Aktivita:

Na základě předcházející diskuse se zavede univerzální třída **Arena**. Definují se další atributy (**spawnPositionX**, **spawnPositionY** a **spawnRotation**), které jsou inicializované v konstruktoru a použité v metodách **spawn()** a **respawn(Enemy)**. Rozměry arény (šířka a výška) jsou též definované a inicializované v konstruktoru. Protože třída **Arena** slouží jako vzor na definování konkrétních arén, je definovaná jako abstraktní třída.

Commit: [e9844d7d9b5f19969618b469ebc907d0fe3c1357](https://github.com/4FUN/4FUN/commit/e9844d7d9b5f19969618b469ebc907d0fe3c1357)

Na základě identifikované třídy **Arena** je definovaná podtřída **DemoArena**. Definuje se konstruktor **DemoArena**, který volá konstruktor nadřazené třídy, a kód odpovědný za rozložení instancí tříd **Orb**, **Tower** a **Direction** se přesune z konstruktoru třídy **Arena** do konstruktoru třídy **DemoArena**. Nakonec se vytvoří nová instance třídy **DemoArena**. Učitel vysvětlí, že pro aktivaci třídy **DemoArena** je nutné na ni v prostředí Greenfoot kliknout pravým tlačítkem myši a vybrat položku „*new DemoArena()*“.

**Commit:** [6a6569774b5735f453a56c7cb2cdbf19d228eae9](#)

### 3. Úloha 6.13.: Vytvoření vlastních arén (30 min)

#### Cíl:

Definování a inicializace vlastních podtříd třídy **Arena**, zkoumání hierarchie tříd **Arena**.

#### Pojmy na diskusi:

- dědičnost, hierarchie tříd, abstraktní třídy, konstruktory.

#### Aktivita:

Vytvoří se další inovativní podtřídy třídy **Arena**. Kód je možné sdílet s ostatními studenty ve skupině.

### 4. Opakování teorie dědičnosti (20 min)

#### Cíl:

Přehled koncepce a výhod dědičnosti, diskuse o hierarchii tříd a abstraktních třídách, zkoumání klíčového slova **super** a LSP spolu s jejich výhodami a zkoumání příkladů a implementací dědičnosti z reálného života a her.

#### Pojmy na diskusi:

- dědičnost, hierarchie tříd, abstraktní třídy, klíčové slovo **super**, LSP, příklady a implementace dědičnosti z reálného života a her.

#### Aktivita:

Prozkoumá se pojem dědičnosti. Prozkoumají se výhody dědičnosti. Rozebere se hierarchie tříd a její výhody v kontextu koncepce dědičnosti. Probere se pojem abstraktní třídy. Rozebere se klíčové slovo **super** a jeho výhody v kontextu dědičnosti. Opět se prozkoumá LSP a výhody jeho použití v kontextu dědičnosti. Diskutuje se o příkladech dědičnosti z reálného života. diskutuje se o příkladech dědičnosti ve hrách a jejich implementací.

## 9. Zapouzdření

V rámci tematického celku zapouzdření byli vytvořené dva metodické materiály.

### 9.1. Zkoumání zapouzdření prostřednictvím vývoje her v prostředí Greenfoot (část 1)

*Tabulka 22. Zkoumání zapouzdření prostřednictvím vývoje her v prostředí Greenfoot (část 1)*

<b>Název</b>	Zkoumání zapouzdření prostřednictvím vývoje her v prostředí Greenfoot (část 1)
<b>Cíle vzdělávání</b>	Studenti porozumějí pojmu zapouzdření prostřednictvím dalšího vývoje hry TowerDefense.
<b>Cílová</b>	Studenti středních škol, kteří se účastní kurzu OOP4Fun. Základní znalosti programování

<i>skupina</i>	včetně dědičnosti. Studenti by se měli seznámit se systémem Greenfoot
<i>Trvání scénáře</i>	<ol style="list-style-type: none"> <li>1. Úvod (5 min) .....</li> <li>2. Úloha 7.1.: Týmová spolupráce a programování (20 min) .....</li> <li>3. Úloha 7.2 a 7.3: Týmová spolupráce (30 min) .....</li> <li>4. Diskuse (35 min) .....</li> <li>5. Vysvětlení kódu (25 min) .....</li> <li>6. Úloha 7.4.: Vytvoření týmu a zadání projektu (10 min) .....</li> </ol>
<i>Materiály a zdroje</i>	<ul style="list-style-type: none"> <li>• učebnice z projektu OOP4Fun</li> <li>• zdroje z projektu OOP4Fun.</li> <li>• zdrojový kód projektu z repozitáře Github/Gitlab.</li> <li>• internetové zdroje.</li> </ul>
<i>Popis</i>	<p>V rámci tohoto 125-minutového výukového scénáře se studenti středních škol dozvědí o zapouzdření v rámci objektově orientovaného programování v prostředí Greenfoot. Lekce začíná stručným 5-minutovým úvodem, ve kterém učitel načrtne cíle.</p> <p>Studenti začnou vývojem třídy <b>ManualTower</b> jako podtřídy třídy Tower. Tato aktivita se zaměří na definování dvou konstruktorů, které budou v souladu s konstruktory rodičovské třídy, aby se zabezpečila správná inicializace. Měli by implementovat metodu <b>act()</b>, která nejdříve zavolá metodu <b>act()</b> rodičovské třídy.</p> <p>Po vytvoření této třídy učitel zavede atribut typu boolean <b>isManuallyControlled</b>, který inicializuje na hodnotu false. Studenti vytvoří metodu <b>changeControl(boolean)</b>, která přepíná stav <b>isManuallyControlled</b> a podle toho mění obraz věže, čímž demonstrují zapouzdření řízením přístupu ke stavu objektu prostřednictvím metod. Každý student by potom měl ručně spustit metodu <b>changeControl()</b> na instancích objektu <b>ManualTower</b> a pozorovat, jak se změní vnitřní jeho stav a zobrazení v aplikaci.</p> <p>Jádrem lekce je vytvoření soukromé metody <b>processUserControl()</b>, která by měla detekovat kliknutí myši na instanci věže. Po kliknutí metoda změní stav ovládání věže a aktualizuje její orientaci na základě polohy myši, přičemž použije zapouzdření na skrytí složité logiky ovládání. Studenti by měli implementovat metodu a integrovat ji do metody <b>act()</b>, testovat interakci s herním prostředím, aby se zabezpečila funkčnost, a naučit se, jakým způsobem chránit kód soukromé metody před vnějšími změnami.</p>
<i>Hodnocení</i>	<p>Táto aktivita umožní učitelům poskytovat formativní zpětnou vazbu na základě diskusí a monitorování obrácení třídy a týmové práce studentů.</p> <p>Vzájemné hodnocení se bude vykonávat online jako součást domácí úlohy. Studentům se tak připomenou důležité aspekty cvičení, donutí se kriticky posoudit práci ostatních studentů, získají přehled o dobrých či méně dobrých řešeních svých spolužáků a zvýší se celkové dosáhnutí výsledků vzdělávání. Při práci na týmové projektu, na kterém studenti pracují, se budou využívat také tyto výsledky vzdělávání a poznatky.</p>
<i>Šíření výsledků</i>	<p>Na šíření výsledků mezi učitelem a studenty se použije repozitář Github/Gitlab a systém řízení výuky (např. Moodle). Studenti mohou pokračovat v diskusi na dané téma na fóru, které je jim</p>

poskytnuté prostřednictvím nástroje na řízení výuky.
--

### 9.1.1. Příručka učitele na přípravu vyučovací hodiny

#### 1. Úvod (5 min)

Učitel by měl spustit dříve vytvořenou hru a nechat studenty pozorovat, jakým způsobem se chovají různé instance. Dále by mělo dojít k návrhu na vytvoření dalšího typu věže, kterou bude možné ovládat ručně, aby se snadněji odstraňovali nepřátelé. Uživatel by měl mít možnost ovládat vždy jednu věž. Po kliknutí myši na věž by se měla stát ručně ovládanou. Na označení toho, která věž je manuálně ovládaná, by měla mít aktuálně ručně ovládaná věž jiný vzhled.

#### 2. Úloha 7.1.: Týmová spolupráce a programování (20 min)

##### Cíl:

Studenti si připraví třídy **ManualTower**.

##### Pojmy na diskusi:

- dědičnost, třídy, konstruktory.

##### Aktivita:

Protože studenti již vědí, jakým způsobem vytvořit třídu potomka, nechte je vytvořit týmy a vytvořit třídu **ManualTower** jako potomka třídy **Tower**. Studenti by měli implementovat konstruktory i metodu **act()** a zabezpečit, aby se z těchto metod volaly konstruktory nadřazené třídy. V této části hodiny si studenti zopakují probranou látku, použijí ji a zdokonalí své praktické znalosti o dědičnost.

**Commit:** [63a02fa0c5080165cba8b467da08c4b65f31d0a8](https://github.com/63a02fa0c5080165cba8b467da08c4b65f31d0a8)

#### 3. Úloha 7.2 a 7.3: Týmová spolupráce (30 min)

##### Cíl:

Učitel vysvětlí studentům nutnost použití soukromých metod v rámci definování funkce **changeControl()**.

##### Pojmy na diskusi:

- metody, třídy, atributy, modifikátory přístupu.

##### Aktivita:

Učitel by měl připravit ikony pro ručně ovládanou věž. Na programovou změnu ikony objektu by měl studentům učitel vysvětlit, jakým způsobem se používá metoda **Actor.setImage(String)**. Studentům nechte určitý čas na vyzkoušení této funkce.

Učitel by měl se studenty prodiskutovat, jakým způsobem lze zjistit, zda je věž ovládaná ručně. Zdůrazněte, že není důležité jen změnit stav objektu, ale také aktualizovat jeho obraz. Zdůrazněte, že pokud chce uživatel změnit stav objektu věže a změní přímo jen atribut, obrázek zůstane stejný. Tato diskuse by měla studentům pomoci pochopit potřebu měnit hodnotu atributu prostřednictvím metody a udržovat atributy spíše jako soukromé než jako veřejné. Studentům vysvětlete, že tento postup se nazývá zapouzdření a že při jeho použití je vnitřní stav skrytý a veřejné metody se používají na kontrolovanou změnu tohoto stavu.

Umožněte studentům implementovat logiku funkce. Nechte je ručně zavolat jejich metodu a pozorovat změny vnitřního stavu.

Commit: [2257746b7dac5eaab7acc55d6493319230338f3a](#)

#### 4. Diskuse (35 min)

##### Cíl:

Porozumění logiky zapouzdření uvnitř samostatné metody.

##### Pojmy na diskusi:

- metody, větvení.

##### Aktivita:

Učitel by měl poukázat na to, že stav věže se nyní dá změnit jen ručním vyvoláním metody. Dále by měl vysvětlit, že myš se může nacházet mimo svět. V takovém případě bude informace o myši mít hodnotu **null**. Studentům připomeňte, že metoda **act()** běží v průběhu hry neustále a že by měla kontrolovat, zda bylo na objekt kliknuto, a až potom zavolat metodu **changeControl()**. Zdůrazněte, že logika zpracování ovládacího prvku by měla být zapouzdřená vevnitř samostatné metody **processUserControl()**.

#### 5. Vysvětlení kódu (25 min)

##### Cíl:

Zavedení metody potřebné na vyřešení problému.

##### Pojmy na diskusi:

- metody, prostředí Greenfoot.

##### Aktivita:

Zvažte, jakým způsobem, změnit stav instance kliknutím na objekt. Na implementaci tohoto postupu je potřebné vysvětlit metodu **GreenFoot.mouseClicked(Object)**. Také představte objekt **MouseInfo**, který je možné použít na získávání informací o poloze myši.

#### 6. Úloha 7.4.: Vytvoření týmu a zadání projektu (10 min)

##### Cíl:

Zlepšení pochopení soukromých metod a zapouzdření prostřednictvím praktického zadání.

##### Pojmy na diskusi:

- metody, modifikátory přístupu, třídy.

##### Aktivita:

Po definování soukromé metody **processUserControl()** nechte studenty implementovat její logiku. Po kliknutí myši by se měla změnit ručně ovládaná věž. Pokud je věž ovládaná ručně, měla být natočena směrem k myši. Připomeňte studentům, že je možné, aby myš byla mimo svět. Po rozdělení studentů do skupin je nechte implementovat logiku metody **processUserControl()**.

Jeden nebo dva týmy odprezentují svoji práci a skupina bude diskutovat o výsledcích spolu s učitelem. Na konci hodiny by měli všichni studenti pochopit, jak se tato metoda implementuje.

Commit: [6ec1f489576019a6493490f9e97797920b923869](#)

## 9.2. Zkoumání zapouzdření prostřednictvím vývoje her v prostředí Greenfoot (část 2)

Tabulka 23. Zkoumání zapouzdření prostřednictvím vývoje her v prostředí Greenfoot (část 2)

<i>Název</i>	Zkoumání zapouzdření prostřednictvím vývoje her v prostředí Greenfoot (část 2)
<i>Cíle vzdělávání</i>	Studenti porozumějí zapouzdření prostřednictvím dalšího vývoje hry TowerDefense.
<i>Cílová skupina</i>	Studenti středních škol, kteří se účastní kurzu OOP4Fun. Základní znalosti programování včetně dědičnosti. Studenti by se měli seznámit se systémem Greenfoot.
<i>Trvání scénáře</i>	<ol style="list-style-type: none"> <li>1. Obrácená třída (30 min) .....</li> <li>2. Atributy třídy (5 min).....</li> <li>3. Úloha 7.6.: Přidat evidenci o ručně řízené věže (5 min).....</li> <li>4. Metody třídy (10 min) .....</li> <li>5. Úloha 7.7: Změna ručně ovládané věže z centrálního místa (20 min).....</li> <li>6. Úloha 7.8: Vyvolání změny ručně ovládané věže (15 min).....</li> <li>7. Opakování teorie (10 min).....</li> </ol>
<i>Materiály a zdroje</i>	<ul style="list-style-type: none"> <li>• učebnice z projektu OOP4Fun</li> <li>• zdroje z projektu OOP4Fun.</li> <li>• zdrojový kód projektu z repozitáře Github/Gitlab.</li> <li>• internetové zdroje.</li> </ul>
<i>Popis</i>	<p>Na začátku hodiny studenti diskutují o problémech souvisejících s ovládáním uživatele, jako je například nemožnost zrušit výběr manuálně ovládané věže po její výběru, a hledají je. Studenti by se měli zapojit do diskuse a navrhnout řešení na sledování aktuálně manuálně ovládané věže a upravit třídu <b>ManualTower</b> tak, aby obsahovala mechanismus na zrušení výběru věže.</p> <p>Nakonec by se měla implementovat metoda třídy <b>changeControlledInstance()</b>, která umožňuje změnu ovládání věží z centralizované metody, čímž se zlepší pochopení zapouzdření tím, že se ukáže, jakým způsobem mohou metody třídy spravovat společný stav mezi instancemi.</p> <p>Tento komplexní vzdělávací přístup vyučuje koncept zapouzdření a demonstruje jeho důležitost a užitečnost v reálných aplikacích, čímž u studentů rozvíjí schopnosti řešit problémy a spolupracovat.</p>
<i>Hodnocení</i>	<p>Táto aktivita umožní učitelům poskytovat formativní zpětnou vazbu na základě diskusí a monitorování obrácení třídy a týmové práci studentů.</p> <p>Práce v týmovém projektu, na kterém studenti pracují, bude také využívat tyto výsledky vzdělávání a poznatky.</p>
<i>Šíření výsledků</i>	Na šíření výsledků mezi učitelem a studenty se použije repozitář Github/Gitlab a systém řízení výuky (např. Moodle). Studenti mohou pokračovat v diskusi na dané téma na fóru, které je jim poskytnuté prostřednictvím nástroje na řízení výuky.

## 9.2.1. Příručka učitele na přípravu vyučovací hodiny

### 1. Obrácená třída (30 min)

#### Cíl:

Studenti by měli rozpoznat potřebu inicializace atributu na jednom místě.

#### Pojmy na diskusi:

- atributy třídy.

#### Aktivita:

Na začátku hodiny nechte studenty identifikovat problém s manuálním ovládním věže uživatelem. V současném stavu projektu není možné zrušit výběr věže. Vyzvěte studenty, aby se zamysleli nad tím, jak by se tento problém dal vyřešit. Vysvětlíte, že ve hře by měla být vždy vybraná jen jedna věž. Tato diskuse by měla studenty přivést k myšlence mít v programu jedno místo, které se inicializuje jen jednou je k němu přístup z jiných částí programů, z jiných objektů instancí.

### 2. Atributy třídy (5 min)

#### Cíl:

Představení základů atributů tříd

#### Pojmy na diskusi:

- atributy, třídy.

#### Aktivita:

Vysvětlíte, co jsou atributy třídy: proměnné, které patří samotné třídě, a ne instancím třídy. Dejte tento koncept do souvislosti se scénářem hry, o kterém se hovořilo dříve, kde by se problém mohl vyřešit tím, že by se centrálně spravoval atribut aktuálně manuálně ovládané věže.

### 3. Úloha 7.6.: Přidat evidenci o ručně řízené věži (5 min)

#### Cíl:

Praktické použití atributů tříd a **null**.

#### Pojmy na diskusi:

- atributy tříd, třídy, modifikátory přístupu.

#### Aktivita:

Pokud chcete sledovat, která věž je v hře aktuálně vybraná k manuálnímu ovládní, přidejte do třídy **ManualTower** soukromý statický atribut **controlledInstance** a inicializujte ho na **null**. Statický atribut se vztahuje na celou třídu, ne na instanci třídy. Proto nám definování statické proměnné umožní zjistit, zda byla vybraná věž, a pokud ano, tak která, odkazem na název třídy bez toho, abychom museli přistupovat ke konkrétnímu objektu. Učitel by měl zdůraznit, že pro celou hru existuje jedna manuálně ovládaná věž, tedy jedna instance. Na začátku by měla být inicializovaná na **null**, protože není vybraná žádná věž. Zkontrolujte vnitřní stav třídy. Zde učitel vysvětlí rozdíly mezi statickými a nestatickými atributy. Učitel se studenty diskutuje o výhodách používání statických atributů ve hrách. Učitel by tu měl připomenout i statické metody a diskutovat se studenty o tom, v čem je jejich používání statických výhodné.

Commit: [c4739460bed583d2126de066acc6b1149d022990](#)

#### 4. Metody třídy (10 min)

**Cíl:**

Představení základních vlastností metod třídy.

**Pojmy na diskusi:**

- metody třídy, třída, objekty.

**Aktivita:**

Představte koncept metod třídy, které mohou pracovat s údaji na úrovni třídy. Diskutujte o potřebě metod jako **changeControlledInstance()** na řízení přepínání aktuálně manuálně ovládané věže. Zdůrazněte, že tyto metody lze volat bez toho, abyste potřebovali instanci třídy. Například školní zvonek zvoní pro všechny ve stejném čase, nezáleží na tom, kdo jste, na druhé straně kontrola domácí úlohy studenta vyžaduje informace o konkrétním studentovi.

#### 5. Úloha 7.7: Změna ručně ovládané věže z centrálního místa (20 min)

**Cíl:**

Praktické použití metod třídy

**Pojmy na diskusi:**

metody, metody třídy, atributy třídy.

**Aktivita:**

Učitel by měl přidat metodu **changeControlledInstance()** na změnu manuálně ovládané věže. Parametrem metody bude věž, kterou chce uživatel vybrat. Nejdříve by se mělo zkontrolovat, či je ovládaná manuálně ovládané instance aktuálně vybraná. Pokud je, nemělo by se nic měnit, ale pokud je manuálně ovládaná instance jiná, měli bychom změnit aktuálně manuálně ovládanou instanci (měl by se změnit odkaz na aktuálně manuálně ovládanou instanci). Funkci vyzkoušejte ručně a pozorujte, že ikony věží se nemění. Poukažte na to, že pouze změna reference na manuálně ovládanou instanci by nezměnila ovládání a že by se to mělo provést ručně. Přidejte kód, který uvolní aktuálně manuálně ovládanou instanci, a po aktualizaci reference přidejte kód, který nastaví manuální ovládání nově manuálně ovládané instance. Zdůrazněte potřebu kontroly nulových referencí, které by se mohly objevit, pokud neexistuje aktuálně manuálně ovládaná instance a pokud neexistuje nově manuálně ovládaná instance (když je parametr nulový).

Commit: [9dc6d8dd4dcbbd71edb8009c1a72403dea1a0ee0](#)

#### 6. Úloha 7.8: Vytvoření změny ručně ovládané věže (15 min)

**Cíl:**

Praktické použití metod třídy.

**Pojmy na diskusi:**

- metody, metody třídy, atributy třídy.

**Aktivita:**

Ručně otestujte, zda se naposledy vytvořená funkce chová správně. Potom se studenty prodiskutujte, kde by se měla tato funkce zavolat. Metoda by měla být vyvolaná vevnitř metody **act()** v třídě **Arena** a vevnitř



metody **changeControl()**. Nakonec zapouzdřete metodu **ManualTower.changeControl(Boolean)** jako soukromou a pozorujte změny instance **ManualTower**.

**Commit:** [c052bbb6aa4c7e690d4d8cf55d3831028fa2b9e3](https://github.com/4FUN/8P/commit/c052bbb6aa4c7e690d4d8cf55d3831028fa2b9e3)

#### 7. Opakování teorie (10 min)

Na závěr provedte souhrn této kapitoly a zdůrazněte důležitost atributů a metod tříd při efektivním řízení herní logiky. Povzbuďte studenty k dalšímu zkoumání aplikováním těchto konceptů ve vlastních programátorských projektech.

## IMPRESUM

*První vydání*

### Název

OOP4FUN: OBJECT ORIENTED PROGRAMMING FOR FUN – Průvodce pro současné učitele středních škol k výuce programování

### Editoři:

Michal Varga, Josef Rak, Dušan Savić, Zlatko Stapić

### Autoři:

Peter Sedláček, Nika Kvaššayová, Jozef Kostolný, Michal Mrena, Patrik Rusnák, Peter Sobe, Ilija Antović, Miloš Milić, Tatjana Stojanović, Davor Fodrek, Lidija Kozina, Marko Mijač, Dijana Plantak Vukovac, Antonela Čižmešija, Dijana Peras, Goran Hajdin, Lea Masnec

### Vydavatel:

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky  
náměstí Čs. legií 565  
530 02 Pardubice,  
Česká republika

### Pro vydavatele:

XX

### Korektor:

XX

### Překlad z anglického jazyka:

Josef Rak

### Grafický design:

Dora Jovanovska, mag. inf.

### Tisk:

powerprint s.r.o.  
Brandejsovo nám. 1219/1,  
165 00 Praha Suchdol

### Náklady:

180

### Místo a rok publikace:

Pardubice, září 2024.

### Podpora:

Financováno Evropskou unií. Vyjádřené názory a názory jsou však pouze názory autora (autorů) a nemusí nutně odrážet názory a názory Evropské unie. Evropská unie ani orgán poskytující podporu za ně nemohou nést odpovědnost