



PRÍRUČKA PRE UČITEĽOV STREDNÝCH ŠKÔL



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

Projekt	Object Oriented Programming for Fun Zábavné objektovo orientované programovanie
Akronym projektu	OOP4FUN
Číslo projektu	2021-1-SK01-KA220-SCH-00027903
Projektový koordinátor	Žilinská univerzita v Žiline (Slovenská republika)
Projektoví partneri	Sveučilište u Zagrebu (Chorvátsko) Srednja škola Ivanec (Chorvátsko) Univerzita Pardubice (Česká republika) Gymnázium, Pardubice, Dašická 1083 (Česká republika) Obchodná akadémia Považská Bystrica (Slovenská republika) Hochschule fuer Technik und Wirtschaft Dresden (Nemecko) Gymnasium Dresden-Plauen (Nemecko) Univerzitet u Beogradu (Srbsko) Gimnazija Ivanjica (Srbsko)
Dátum vydania	2024

Vyhlásenie o vylúčení zodpovednosti:

Financované Európskou úniou. Vyjadrené názory a postoje sú názormi a vyhláseniami autorov a nemusia nevyhnutne odrážať názory a stanoviská Európskej únie alebo Slovenská akademická asociácia pre medzinárodnú spoluprácu. Európska únia ani Slovenská akademická asociácia pre medzinárodnú spoluprácu za ne nepreberajú žiadnu zodpovednosť.

Obsah

Obsah.....	5
Zoznam tabuliek	7
1. Informačný prehľad	9
2. Úvod do vývojového prostredia Greenfoot	13
2.1. Tvorivé skúmanie vývoja hier	13
2.1.2. Príručka učiteľa na prípravu vyučovacej hodiny	15
3. Definícia triedy	19
3.1. Skúmanie tried a objektov prostredníctvom vývoja hier s Greenfoot.....	19
3.1.2. Príručka učiteľa na prípravu vyučovacej hodiny	20
3.2. Vytváranie tried a objektov prostredníctvom vývoja hier s Greenfoot	24
3.2.2. Príručka učiteľa na prípravu vyučovacej hodiny	25
4. Algoritmus	29
4.1. Úvod do algoritmov v prostredí Greenfoot	29
4.1.2. Príručka učiteľa na prípravu vyučovacej hodiny	30
4.2. Greenfoot dobrodružstvá: Volanie metód v Java, práca s dokumentáciou a riadenie aplikácie.....	33
4.2.2. Príručka učiteľa na prípravu vyučovacej hodiny	35
5. Vetvenie.....	41
5.1. Skúmanie vetvenia pri vývoji hier s Greenfoot - Neúplné vetvenie kódu	41
5.1.2. Príručka učiteľa na prípravu vyučovacej hodiny	42
5.2. Skúmanie vetvenia pri vývoji hier s Greenfoot - Úplné vetvenie kódu	45
5.2.2. Príručka učiteľa na prípravu vyučovacej hodiny	47
6. Premenné a výrazy	51
6.1. Úvod do premenných a dátových typov v prostredí Greenfoot.....	51
6.1.1. Príručka učiteľa na prípravu vyučovacej hodiny	52
6.2. Úvod do operátorov a výrazov v prostredí Greenfoot	54
6.2.1. Príručka učiteľa na prípravu vyučovacej hodiny	56
6.3. Úvod do konštruktorov v prostredí Greenfoot.....	59
6.3.1. Príručka učiteľa na prípravu vyučovacej hodiny	60
6.4. Úvod do atribútov v prostredí Greenfoot.....	62

6.4.1.	Príručka učiteľa na prípravu vyučovacej hodiny	63
6.5.	Úvod do preťažovania konštruktorov v prostredí Greenfoot	65
6.5.1.	Príručka učiteľa na prípravu vyučovacej hodiny	66
7.	Asociácia	69
7.1.	Greenfoot objekty a ich spolupráca: preskúmanie metód a asociácií	69
7.1.1.	Príručka učiteľa na prípravu vyučovacej hodiny	71
7.2.	Greenfoot objekty a ich spolupráca: preskúmanie asociácií a pokročilých volaní metód 75	
7.2.1.	Príručka učiteľa na prípravu vyučovacej hodiny	77
7.3.	Greenfoot objekty a ich spolupráca: Veže, strely a strategické interakcie	81
7.3.1.	Príručka učiteľa na prípravu vyučovacej hodiny	83
7.4.	Greenfoot objekty a ich spolupráca: strely, nepriatelia a dynamika hry	88
7.4.1.	Príručka učiteľa na prípravu vyučovacej hodiny	91
8.	Dedičnosť	95
8.1.	Úvod do dedičnosti v prostredí Greenfoot.....	95
8.1.1.	Príručka učiteľa na prípravu vyučovacej hodiny	97
8.2.	Koncepty dedičnosti v prostredí Greenfoot (časť 1)	99
8.2.1.	Príručka učiteľa na prípravu vyučovacej hodiny	101
8.3.	Koncepty dedičnosti v prostredí Greenfoot (časť 2)	103
8.3.1.	Príručka učiteľa na prípravu vyučovacej hodiny	105
8.4.	Koncepty dedičnosti v prostredí Greenfoot (časť 3)	106
8.4.1.	Príručka učiteľa na prípravu vyučovacej hodiny	108
9.	Zapuzdrenie.....	111
9.1.	Skúmanie zapuzdrenia prostredníctvom vývoja hier s Greenfoot	111
9.1.1.	Príručka učiteľa na prípravu vyučovacej hodiny	112
9.2.	Skúmanie zapuzdrenia prostredníctvom vývoja hier s Greenfoot	115
9.2.1.	Príručka učiteľa na prípravu vyučovacej hodiny	116

Zoznam tabuliek

Tabuľka 1: Šablóna na dokumentovanie metodických materiálov	10
Tabuľka 2: Tvorivé skúmanie vývoja hier	13
Tabuľka 3: Skúmanie tried a objektov prostredníctvom vývoja hier s Greenfoot	19
Tabuľka 4: Vytváranie tried a objektov prostredníctvom vývoja hier s Greenfoot.....	24
Tabuľka 5: Úvod do algoritmov a algoritmického myslenia	29
Tabuľka 6: Greenfoot dobrodružstvá: Volanie metód v Java, práca s dokumentáciou a riadenie aplikácie.....	33
Tabuľka 7: Skúmanie vetvenia pri vývoji hier s Greenfoot - Neúplné vetvenie kódu	41
Tabuľka 8: Skúmanie vetvenia pri vývoji hier s Greenfoot - Úplné vetvenie kódu	45
Tabuľka 9: Úvod do premenných a dátových typov v prostredí Greenfoot.....	51
Tabuľka 10: Úvod do operátorov a výrazov v prostredí Greenfoot	54
Tabuľka 11: Úvod do konštruktorov v prostredí Greenfoot.....	59
Tabuľka 12: Úvod do atribútov v prostredí Greenfoot.....	62
Tabuľka 13: Úvod do preťažovania konštruktorov v prostredí Greenfoot.....	65
Tabuľka 14: Greenfoot objekty a ich spolupráca: preskúmanie metód a asociácií	69
Tabuľka 15: Greenfoot objekty a ich spolupráca: preskúmanie asociácií a pokročilých volaní metód	75
Tabuľka 16: Greenfoot objekty a ich spolupráca: Veže, strely a strategické interakcie	81
Tabuľka 17: Greenfoot objekty a ich spolupráca: strely, nepriatelia a dynamika hry	88
Tabuľka 18: Úvod do dedičnosti v prostredí Greenfoot.....	95
Tabuľka 19: Koncepty dedičnosti v prostredí Greenfoot	99
Tabuľka 20: Koncepty dedičnosti v prostredí Greenfoot	103
Tabuľka 21: Koncepty dedičnosti v prostredí Greenfoot	106
Tabuľka 22: Skúmanie zapuzdrenia prostredníctvom vývoja hier s Greenfoot	111
Tabuľka 23: Skúmanie zapuzdrenia prostredníctvom vývoja hier s Greenfoot	115

1. Informačný prehľad

Cieľom tejto knihy je predstaviť kurz s materiálmi, ktoré pomôžu učiteľom pri príprave materiálov na vyučovanie v oblasti riešenia programovacích úloh s využitím základov objektovo orientovaného programovania (OOP) podľa paradigmy jednoduchého OOP.

Žiaci sa naučia rozdeliť zadané úlohy medzi spolupracujúce objekty; určiť ich kompetencie a implementovať navrhnutý model. Predmet nevyžaduje predchádzajúce znalosti programovania. Vyučuje sa v programovacom jazyku Java. Používame vývojové prostredie Greenfoot, ktoré podporuje programovací jazyk Java. Java je v súčasnosti veľmi populárny a v praxi široko používaný programovací jazyk. Greenfoot navyše predstavuje editor zdrojového kódu založený na rámcoch, ktorý využíva jazyk Stride. To otvára možnosti pre učiteľov, ktorí budú chcieť používať v tomto učebnom pláne prezentované techniky so žiakmi mladšieho veku. Greenfoot je veľmi vizuálny a od začiatku umožňuje vytvoriť vizualizovaný objekt, ktorý je „živý“ a dá sa s ním interagovať. Preto je teoretický úvod minimalizovaný a žiaci začnú pracovať hneď od začiatku.

Kurz vysvetľuje jednoduché koncepty OOP (ako je zapúzdrenie, dedičnosť alebo asociácia) na tvorbe počítačových hier, kde sa tieto koncepty jednoducho a intuitívne využívajú. Proces tvorby počítačovej hry je založený na tímovej práci a prakticky využíva vedomosti a zručnosti z iných oblastí informatiky a s ňou súvisiacich predmetov (práca s multimédiami a kancelárskym softvérom). Dizajn každej počítačovej hry je dostatočne otvorený na to, aby žiaci mohli hru individuálne a tvorivo rozšíriť. Okrem toho návrh vedie k správne využitiu získaných vedomostí.

Kniha je zameraná na predstavenie inovatívneho prístupu k výučbe programovania, založeného na riešení úloh s využitím paradigmy objektovo orientovaného programovania (OOP). OOP je v súčasnosti dominantnou paradigmou pre vývoj aplikácií. Preto je vhodné, aby žiaci mali vedomosti a zručnosti z tejto oblasti. Predmet predstavuje vývojové prostredie, ktoré využíva rôzne formy úpravy zdrojového kódu (rámcová úprava pomocou zjednodušenej formy, ako aj reálne písanie zdrojového kódu), čo umožňuje vyučovať žiakov na rôznych úrovniach predchádzajúcich technických znalostí a aktivít. Svojou jednoduchosťou a prehľadnosťou tento nástroj podporuje rýchle a intuitívne pochopenie vyučovaných tém, čo má pozitívny vplyv na žiakov a ich motiváciu.

Prostredníctvom programovania interaktívnych hier v grafickom prostredí žiak získava vedomosti a zručnosti, takže bude schopný:

- identifikovať problém,
- identifikovať vhodné objekty na riešenie identifikovaného problému (dekompozícia objektov),
- navrhnuť triedy objektov, ako aj ich atribúty a metódy,
- identifikovať a správne využívať vzťahy medzi objektmi (asociácia, dedičnosť),
- navrhnuť algoritmus na riešenie problému a rozdeliť ho medzi kooperujúce objekty,

- používať prvky zdrojového kódu (vetvenie, cykly) na implementáciu navrhnutého algoritmu,
- efektívne používať prostriedky na debugovanie (ladenie) zdrojového kódu,
- vytvoriť jednoduchú aplikáciu s grafickým rozhraním vo vývojovom prostredí Greenfoot.

Výsledky vzdelávania z predmetu sú zhrnuté nasledovne:

- pochopenie základných princípov objektovo orientovaného programovania,
- pochopenie základov algoritmizácie,
- pochopenie syntaxe programovacieho jazyka Java,
- analýza vykonávania programu na základe zdrojového kódu,
- schopnosť vytvárať vlastné programy s využitím OOP.

Moderným prístupom pri koncipovaní prednášok, najmä tých pre základné a stredné školy, je definovanie a zdieľanie metodických . Metodické materiály (MM) „sú vnímané ako moderný pedagogický prístup, ktorý umožňuje individualizáciu vyučovacieho procesu zohľadnením rôznych potrieb žiakov. Vyučovanie založené na MM sa zameriava na relevantné vedomosti a zručnosti žiakov vrátane tých, ktoré sú potrebné pre digitálnu spoločnosť. Starostlivé plánovanie MM môže odstrániť možné úskalia a nedostatky, ktoré by mohli ovplyvniť vyučovací proces.“

V kontexte vzdelávania a navrhovania vyučovania predstavujú MM podrobné opisy alebo rozprávanie, ktoré načrtávajú konkrétnu vyučovaciu situáciu alebo kontext. Tieto MM sa často používajú pri vzdelávaní učiteľov na simuláciu reálnych vyučovacích situácií, a preto ich považujeme za najlepší nástroj na prezentáciu našich inovatívnych nápadov v oblasti vyučovania a učenia sa. Keďže metodické materiály zvyčajne obsahujú informácie o cieľoch vyučovania, obsahu, ktorý sa má vyučovať, charakteristikách učiacich sa, použitých vyučovacích metódach a stratégiách hodnotenia, možno ich zosúladiť aj s prvkami potrebnými pre naše vzdelávacie tematické plány.

Aby sme mali štruktúrovaný prístup pri definovaní niekoľkých MM, definovali sme nasledujúcu šablónu, ktorá sa naplňuje konkrétnymi informáciami týkajúcimi sa konkrétnej témy. Šablóna obsahuje krátky popis toho, ako definovať jednotlivé prvky MM.

Tabuľka 1: Šablóna na dokumentovanie metodických materiálov

<p><i>Názov</i></p>	<p>Pridať opisný a pútavý názov.</p>
<p><i>Výchovno- vzdelávacie ciele a ich opis</i></p>	<p>Jasne uviesť plánované výsledky vzdelávania. Čo by mali žiaci vedieť, čomu by mali rozumieť alebo čo by mali byť schopní urobiť po zvládnutí metodiky?</p>

<i>Cieľová skupina</i>	Uviesť cieľovú skupinu, ročník a už existujúce vedomosti, pre ktoré je metodický materiál (MM) určený.
<i>Časové trvanie</i>	Odhadnúť čas potrebný na dokončenie MM vrátane všetkých konkrétnych časových rámcov pre rôzne aktivity. Napr. motivácia (5 min), bádanie, ktoré vykonajú žiaci samostatne (10 min), programovanie riešenia v tíme (20 min), prezentácia/diskusia (10 min.).
<i>Materiály a zdroje</i>	Uviesť zoznam materiálov, zdrojov a nástrojov potrebných pre učiteľov aj žiakov. Môže to zahŕňať učebnice, online a multimediálne zdroje, softvér atď.
<i>Opis</i>	<ul style="list-style-type: none"> • Predstavte cieľ vyučovania, vysvetlite jeho účel a význam. • Načrtnite základné aktivity/úlohy, ktoré budú žiaci vykonávať na dosiahnutie cieľov vzdelávania. Uveďte podrobnosti, ako sú diskusie, praktické činnosti, skupinová práca, súťaž atď. • Uveďte, ako budú žiaci organizovaní, t. j. či budú pracovať individuálne alebo v tíme. Aké veľké budú tímy? • Vysvetlite, na akých projektoch/problémoch/úlohách budú žiaci pracovať. Odporúčame použiť problémový alebo projektový prístup. Aj tie by mali odrážať situácie z reálneho života. • Vysvetlite, ako budú projekty/problémy/úlohy pridelené žiakom (tímom). • Ak pracujú v tímoch, uveďte podrobnosti o tom, ako budú spolupracovať. • Uveďte viac podrobností týkajúcich sa aktivít, na ktorých sa žiaci musia zúčastniť. • Ak sa používa napr. obrátený spôsob výučby, uveďte, akú časť danej témy musia žiaci sami preskúmať.
<i>Hodnotenie</i>	<p>Uviesť podrobnosti týkajúce sa toho, ako sa bude hodnotiť úsilie a vedomosti žiakov.</p> <ul style="list-style-type: none"> • Kto bude žiakov hodnotiť: (1) učitelia, (2) žiaci svoju vlastnú prácu (sebahodnotenie), (3) žiaci navzájom svoju prácu (vzájomné hodnotenie). • Aké kritériá hodnotenia sa budú používať? • Ako často sa bude hodnotenie vykonávať?

	<ul style="list-style-type: none">• atď.
<i>Šírenie výsledkov</i>	Vysvetlite, ako budú žiaci šíriť svoje výsledky učiteľom a spolužiakom. Napr. Žiaci (všetci alebo podskupina) môžu prezentovať svoje výsledky/riešenia pred učiteľom a svojimi spolužiakmi, a potom môže nasledovať porovnanie a diskusia.

V súlade s tým nájdete náš tematický plán na tejto webovej adrese: Podrobnosti o kurze (learning-design.eu). Materiály, ktoré podporujú túto knihu, nájdete na platforme Moodle: <https://oop4fun.fon.bg.ac.rs/>.

2. Úvod do vývojového prostredia Greenfoot

Greenfoot je vizuálny 2D vzdelávací softvérový nástroj s editorom programu na vytváranie hier a simulácií v programovacom jazyku Java. Greenfoot je vizuálny a interaktívny. Programuje sa v štandardnom textovom programovacom jazyku Java, čo poskytuje kombináciu skúseností s programovaním v tradičnom textovom jazyku s vizuálnym vykonávaním.

2.1. Tvorivé skúmanie vývoja hier

Tabuľka 2: Tvorivé skúmanie vývoja hier

Názov	Tvorivé skúmanie vývoja hier
Výchovno-vzdelávacie ciele a ich opis	Na konci tohto stretnutia budú mať žiaci nielen úspešne nainštalovaný Greenfoot a overené jeho možnosti na príkladoch projektov , ale budú sa tiež zapájať do spoločnej praktickej hry vo vývojovom prostredí. Tento hravý úvod udáva tón vzrušujúcemu skúmaniu vývoja hier, podporuje tvorivosť, tímovú prácu a nadšený prístup k programovaniu s Greenfoot.
Cieľová skupina	Žiaci stredných škôl, ktorí sa zúčastňujú kurzu OOP4Fun. Základné znalosti programovania vrátane premenných, funkcií, iterácií a vetvenia.
Časové trvanie	<ol style="list-style-type: none">1. Úvod (5 min)2. Rýchla výzva (10 min)3. Hranie hier s učiteľom (30 min)4. Tvorba tímu a zadanie projektu (5 min)5. Tímová spolupráca a programovanie (30 min)6. Vzájomné hodnotenie a spätná väzba (10 min)7. Domáce úlohy (30 min)8. Hodnotenie súťaže (30 min)
Materiály a zdroje	<ul style="list-style-type: none">• Webová stránka Greenfoot a pokyny na stiahnutie.• Príklady pripravené učiteľom. Internetové zdroje na identifikáciu ďalších príkladov.
Opis	V tomto 90-minútovom metodickom materiáli sa žiaci stredných škôl ponoria do sveta Greenfoot prostredníctvom hry, zábavy,

	<p>výskumu/bádania a tímovej práce.</p> <p>Po tom, ako učiteľ predstaví dnešnú lekciu, zreflektuje predchádzajúcu lekciu a stanoví ciele, začína sa rýchla výzva. Žiaci dostanú gamifikovanú úlohu nájsť návod, stiahnuť a nainštalovať Greenfoot (pre nich zatiaľ neznámy vývojový nástroj) do svojich počítačov. Prví traja žiaci dostanú žetóny ocenenia (odznaky, body, skóre, sladkosti atď.).</p> <p>Druhým prekvapením pre nich je, že v nasledujúcich 30 minútach budú hrať hry s učiteľom. Ide o stretnutie pod vedením učiteľa zamerané na otvorenie, kompiláciu a spustenie jedného-dvoch jednoduchých príkladov projektov (na úvodnej až strednej úrovni zložitosti). Žiaci sa tak oboznámia so základnými prvkami vývojového prostredia Greenfoot a so základnými postupmi pri práci so súbormi a vlastnosťami projektu.</p> <p>Potom budú žiaci rozdelení do tímov (po 3 - 4 žiakoch) a dostanú jednoduché zadanie. Tímy by mali „niečo“ v danom ukážkovom projekte zmeniť, aby hra bola prekvapivá alebo zábavná. V rámci tímovej spolupráce a programovania (30 minút) budú tímy spoločne pracovať na tom, aby sa pokúsili niečo zmeniť v daných príkladoch. Ak porušia program za hranicou toho, že ho dokážu opraviť sami, môžu požiadať o pomoc učiteľa alebo si môžu znovu stiahnuť „štartovaciu verziu“. Toto bude dobrý príklad, prečo by sme mali pri programovaní používať systémy kontroly verzií.</p> <p>Jeden alebo dva tímy predstavia svoju prácu na vzájomné posúdenie a spätnú väzbu a prediskutujú výsledky s učiteľom.</p> <p>Doma na domácu úlohu by mal každý žiak vyhľadať príklady hier Greenfoot a mal by svojej triede predstaviť svoj obľúbený príklad tak, že nahrá internetový odkaz, opis toho, čo robí tento príklad jeho obľúbeným, a dva-tri screenshoty vývojového prostredia a spustenej hry. V rámci gamifikácie a motivačie prostredníctvom súťaže by mal každý žiak hlasovať za tri najlepšie hry (nie je dovolené hlasovať za vlastnú hru). Víťazi sú vyhlásení a odmenení žetónmi uznania (odznaky, body, skóre, sladkosti atď.).</p>
<p><i>Hodnotenie</i></p>	<p>Táto aktivita umožní učiteľom poskytovať formatívnu spätnú väzbu na základe diskusií a monitorovania obrátenej triedy a tímovej práce žiakov.</p> <p>Gamifikácia predstavuje neformálne hodnotenie, ale zvýši záujem, vnútornú motiváciu a výstupy z učenia celej skupiny. Hodnotenie vzájomného hodnotenia sa bude vykonávať online ako súčasť domácej úlohy. To žiakom pripomenie dôležité aspekty hodiny, prinúti ich nainštalovať Greenfoot, uvedie rôzne príklady a pripomenie im, čo sa</p>

	počas hodiny robilo, čo zvýši celkové dosiahnutie výsledkov vzdelávania.
Šírenie výsledkov	Na šírenie výsledkov medzi učiteľmi a spolužiakmi sa použije bežné nastavenie Learning management systému (napr. Moodle). Žiaci môžu pokračovať v diskusii na danú tému na fóre, ktoré im je poskytnuté prostredníctvom nástroja na Learning management.

2.1.2. Príručka učiteľa na prípravu vyučovacej hodiny

1. Úvod

Cieľ:

Žiaci sa oboznámia s programovacími jazykmi (vizuálnymi a textovými), integrovanými vývojovými prostrediami a zdrojovým kódom.

Pojmy na diskusiu:

Programovacie jazyky, integrované vývojové prostredia, zdrojový kód.

Aktivita:

Učiteľ urobí krátky úvod k dnešnej hodine. Učiteľ predstaví pojmy ako:

- Programovací jazyk,
- integrované vývojové prostredia (programovacie nástroje) a
- zdrojový kód.

Učiteľ prezentuje príklady zdrojových kódov v rôznych programovacích jazykoch, ako sú Java, C a Python, ale predvádza aj programy v jazyku Scratch alebo inom vizuálnom jazyku. Učiteľ nejde "do hĺbky", ale snaží sa tieto pojmy vysvetliť na jednoduchom príklade, napríklad sa snaží porovnať učenie sa programovacieho jazyka s učením sa rodného alebo cudzieho jazyka.

- Prirodzené jazyky majú svoju vlastnú gramatiku a pravopis, rovnako je to aj v prípade programovacích jazykov. Pri písaní sa riadime určitými gramatickými a pravopisnými pravidlami, podobne aj pri písaní programu v programovacom jazyku.
- Keď píšeme príbeh v prirodzenom jazyku, používame zošit, pero ako nástroj, ktorý nám pri tom pomáha, podobne keď píšeme program v programovacom jazyku, naším "nástrojom" je integrované vývojové prostredie.
- Výsledkom písania môže byť príbeh napísaný na papieri, zatiaľ čo výsledkom programovania je program, ktorý vytvoríme a ktorý označujeme ako zdrojový kód.
- Učiteľ môže urobiť úvod na vlastnom príklade.

2. Rýchla výzva

Cieľ:

Žiaci preskúmajú Greenfoot a návod na inštaláciu.

Pojmy na diskusiu:

Greenfoot, návod na inštaláciu.

Aktivita:

Učiteľ zadá žiakom úlohu zistiť, čo je to Greenfoot. Učiteľ povie žiakom, aby našli návod na inštaláciu Greenfootu. Žiaci pracujú na úlohe, potom učiteľ žiakom predstaví, ako nájsť, stiahnuť a spustiť návod na inštaláciu Greenfootu. Návod na stiahnutie a inštaláciu Greenfootu učiteľ nájde na platforme Moodle.

3. Hranie hier s učiteľom

Cieľ:

Žiaci spustia rôzne projekty Greenfoot.

Pojmy na diskusiu:

Greenfoot projekt (webové a samostatné projekty).

Aktivita:

Učiteľ zadá žiakom úlohu, aby si na internete pozreli a našli príklady projektov vytvorených v Greenfoot. Žiaci hľadajú projekty, zatiaľ čo učiteľ sleduje ich prácu.

Učiteľ predstaví žiakom, ako môžu nájsť príklady hotových projektov vytvorených v prostredí Greenfoot. Napríklad vo vyhľadávači Google možno nájsť projekty na základe kľúčových slov "Greenfoot Java project example" alebo podobne.

Učiteľ vysvetlí žiakom, že existujú dva typy Greenfoot projektov:

- jeden, ktorý možno spustiť vo webovom prehliadači, a
- ďalší, ktorý možno stiahnuť a potom otvoriť a spustiť z prostredia Greenfoot.

Učiteľ predstaví projekty:

- projekty, ktoré možno spustiť priamo vo webovom prehliadači,
- projekty, ktoré možno spustiť v Greenfoot po stiahnutí.

Učiteľ si môže stiahnuť niekoľko príkladov projektov alebo sa k niektorým projektom dostať pomocou odkazov.

4. Vytvorenie tímu a zadanie projektu

Cieľ:

Žiak sa zapojí do projektového vyučovania pomocou jednoduchého zadania.

Pojmy na diskusiu:

Aktivita:

Učiteľ vytvorí tímy, pripraví úlohu a zadá projekt, na ktorom budú žiaci pracovať. Niektoré príklady úloh nájdete na Moodle. Učiteľ vyberie jeden reprezentatívny projekt (nie veľmi zložitý) a pre tento projekt definuje úlohu (problém), ktorú majú žiaci vyriešiť.

5. Tímová spolupráca a programovanie

Cieľ:

Žiak vie vytvoriť projekt v prostredí Greenfoot.

Pojmy na diskusiu:

Vytvorenie projektu v prostredí Greenfoot.

Aktivita:

Učiteľ spustí Greenfoot a predstaví žiakom, ako vytvoriť projekt. Učiteľ zadá žiakom úlohu 1.1 - vytvoriť projekt Greenfoot.

Učiteľ zdôrazní žiakom, že projekt (totožný s projektom, ktorý bol práve vyhotovený) môže byť:

- stiahnutý z gitu zadaním nasledujúceho odkazu
Commit: [9046f5353d857dcc112abd92d7b7170abcc64a80](https://github.com/9046f5353d857dcc112abd92d7b7170abcc64a80))
- stiahnutý z gitu, ale súbor ZIP z nasledujúcej adresy
<https://oop4fun.fon.bg.ac.rs/>

Učiteľ zdôrazní žiakom, že na dnešnej hodine nebudú pracovať na projekte, ktorý práve vytvorili, ale až od nasledujúcej hodiny, a že teraz budú pracovať na existujúcich projektoch.

Žiaci sa ujmú úlohy a riešia problém.

6. Vzájomné hodnotenie a spätná väzba

Cieľ:

Žiaci vedia viesť diskusiu o riešení, ktoré je navrhnuté žiakmi.

Pojmy na diskusiu:

Aktivita:

Učiteľ sleduje prácu žiakov a v prípade potreby im dáva pokyny (inštrukcie). Po skončení učiteľ vyberie jeden tím, ktorý ukáže svoje riešenie. Učiteľ diskutuje so žiakmi o navrhovanom riešení.

7. Domáce úlohy

Cieľ:

Žiaci vedia preskúmať projekty Greenfoot.

Pojmy na diskusiu:**Aktivita:**

Učiteľ definuje úlohu, ktorú majú žiaci vyriešiť na jednom z existujúcich projektov.

8. Klasifikácia súťaže

Cieľ:

Žiaci sa zapájajú do procesu hodnotenia ich projektov.

Pojmy na diskusiu:**Aktivita:**

Žiaci prezentujú svoj projekt. Učiteľ vyžaduje, aby sa všetci zúčastnili na hodnotení prezentovaných projektov zaslaním odkazov žiakom na vyplnenie formulára Google. Žiaci vyhodnotia tri najlepšie tímy.

Po prezentáciách učiteľ zhrnie žiacke projekty. Učiteľ sa podelí o svoje dojmy z práce žiakov a uvedie, či je s ňou spokojný, či splnila jeho očakávania, alebo ich prekonal.

3. Definícia triedy

V rámci tematického celku Definícia triedy boli vytvorené dva metodické materiály.

3.1. Skúmanie tried a objektov prostredníctvom vývoja hier s Greenfoot

Tabuľka 3: Skúmanie tried a objektov prostredníctvom vývoja hier s Greenfoot

<i>Názov</i>	Skúmanie tried a objektov prostredníctvom vývoja hier s Greenfoot
<i>Výchovno-vzdelávacie ciele a ich opis</i>	<p>Na konci tejto lekcie budú žiaci schopní pochopiť základný koncept objektu a triedy.</p> <p>Pochopenie skúmaných konceptov sa bude diskutovať v kontexte vývoja hry, čo podporí tvorivosť, tímovú prácu a motivujúci prístup k programovaniu s nástrojom Greenfoot.</p>
<i>Cieľová skupina</i>	Žiaci stredných škôl na kurze OOP4Fun. Základné znalosti programovania vrátane objektov a tried.
<i>Časové trvanie</i>	<ol style="list-style-type: none">1. Objekt (10 min)2. Identifikácia objektov a ich vlastností (15 min)3. Trieda, inštancia, dedičnosť (15 min)4. Orientácia v Greenfoot: (10 min)5. Konštruktor triedy (10 min)6. Úloha 1.2 (15 min)7. Nastavenie obrázkov (10 min)8. Úloha 1.3 (15 min)
<i>Materiály a zdroje</i>	<ul style="list-style-type: none">• Webová stránka Greenfoot a pokyny na stiahnutie.• Príklady pripravené učiteľom.• Internetové zdroje ako ukážku ďalších príkladov.
<i>Opis</i>	<p>V tomto 90-minútovom metodickom materiáli sa žiaci stredných škôl oboznámia s princípmi programovania súvisiacimi s objektmi a triedami z pohľadu vývoja hier pomocou nástroja Greenfoot.</p> <p>Hodina sa začína 10-minútovým úvodom učiteľa, ktorý uvedie žiakov do sveta objektovo orientovaného programovania tým, že vysvetlí pojem</p>

	<p>objektu a jeho vlastností v reálnom živote.</p> <p>Potom učiteľ zadá žiakom úlohu v rýchlej výzve (10 minút), v ktorej musia na základe textového opisu úlohy identifikovať predmety a ich vlastnosti. Potom učiteľ spolu so žiakmi vymyslí riešenie úlohy (5 minút).</p> <p>V pokračovaní hodiny učiteľ vysvetlí rozdiel medzi triedou a predmetom. Na najvyššej úrovni abstrakcie vysvetlí pojem dedičnosť (15 minút).</p> <p>Učiteľ spustí prostredie Greenfoot a vysvetlí triedy World, Actor a MyWorld (10 minút).</p> <p>Učiteľ vysvetlí a prezentuje zdrojový kód, ktorý bol vygenerovaný a stojí za triedami World, Actor, MyWorld po vytvorení projektov (10 minút).</p> <p>Učiteľ spustí úlohu 1.2 a ukáže žiakom, ako vytvoriť World pre aplikáciu, ktorú potrebujú vytvoriť (10 minút).</p> <p>Učiteľ vysvetlí, ako nastaviť obrázok pre konkrétnu triedu a spolu so žiakmi pracuje na úlohe 1.3 (20 minút).</p>
<i>Hodnotenie</i>	Gamifikácia predstavuje neformálne hodnotenie, ale zvýši záujem, vnútornú motiváciu a vzdelávacie výstupy celej skupiny.
<i>Šírenie výsledkov</i>	Na šírenie svojich výsledkov učiteľovi a spolužiakom sa použije zvyčajné nastavenie Github/Gitlab a Learning management system (napr. Moodle). Žiaci môžu pokračovať v diskusii na danú tému na fóre, ktoré im je poskytnuté prostredníctvom nástroja na Learning management.

3.1.2. Príručka učiteľa na prípravu vyučovacej hodiny

1. Objekt

Cieľ:

Žiaci sa oboznámia s pojmom objekt na príkladoch zo života.

Pojmy na diskusiu:

Objekt a vlastnosti objektu.

Aktivita:

Učiteľ zavedie pojem objekt. Učiteľ by mal tento pojem priblížiť žiakom pomocou príkladov z reálneho života. Napríklad by sa mohol spýtať žiakov na ich meno, výšku, dátum narodenia, farbu očí... Učiteľ kladie tieto otázky, aby podnietil žiakov zamyslieť sa nad tým, čím sa od seba líšia, a pripravil ich tak na neskoršiu otázku: Ako sa žiaci od seba navzájom líšia?

Učiteľ konštatuje, že každý z nás (učiteľ, študenti, žiaci) má vlastnosti, ktorými sa od seba navzájom odlišujeme, a zdôrazňuje, že každý z nás je vlastne "jeden" objekt. Učiteľ vysvetlí pojem vlastnosť ako vlastnosť, ktorú má každý z nás, spolu s konkrétnou hodnotou každej vlastnosti, ktorá je jedinečná pre každého jednotlivca. Objekty sa teda od seba navzájom líšia na základe hodnôt, ktoré majú pre svoje príslušné vlastnosti.

2. Identifikácia objektov a ich vlastností

Cieľ:

Žiaci vedia identifikovať predmety a ich vlastnosti.

Pojmy na diskusiu:

Objekty a ich vlastnosti.

Aktivita:

Učiteľ zadá žiakom úlohu, aby na základe textu určili predmety a ich vlastnosti.

3. Trieda, inštancia, dedičnosť

Cieľ:

Žiaci získajú vedomosti o triede a inštancii. Vedia rozlišovať medzi triedou a inštanciou.

Pojmy na diskusiu:

Trieda, inštancia

Aktivita:

Na vysvetlenie pojmu trieda, inštancie triedy (objekty) a dedičnosť učiteľ uvádza príklady z reálneho života. Učiteľ kladie žiakom otázky, aby pochopili rozdiel medzi triedou a objektom. Učiteľom riadená diskusia o rozpoznaných objektoch a ich zaradení do tried.

4. Vytvorenie inštancie sveta v Greenfoot

Cieľ:

Žiaci zvládnu úvod do sveta v Greenfoot.

Pojmy na diskusiu:

Inštancia sveta v Greenfoot.

Aktivita:

Učiteľ spustí prostredie Greenfoot a vytvorí jednoduchý projekt. Učiteľ použije tento projekt na vysvetlenie, ako žiaci vytvárajú zavedený koncept do Greenfoot.

Učiteľ si všimol, že každý projekt vytvorený v prostredí Greenfoot obsahuje 3 triedy: **World**, **Actor** a **MyWorld**.

Učiteľ predstaví triedu **MyWorld** a jej úlohu.

Učiteľ zdôrazní, že pozadie každej aplikácie vytvorenej v prostredí Greenfoot pozostáva z buniek, ktoré predstavujú jednu maticu. Učiteľ ukáže, ako definovať veľkosť pozadia (rozmer matice) a veľkosť každej bunky matice.

Učiteľ vysvetlí, že objekty, ktoré sa zobrazujú v tabuľke („scéna“), sa nachádzajú na jednej z týchto buniek. Učiteľ ukáže zdrojový kód za každou z týchto tried.

5. Príprava sveta

Cieľ:

Žiaci si zapojení do práce na projektovej úlohe.

Pojmy na diskusiu:

Svet v prostredí Greenfoot.

Aktivita:

Učiteľ zadá žiakom úlohu 1.2. Učiteľ sleduje prácu žiakov a na záver požiada jedného žiaka, aby ukázal svoje riešenie a opísal ho.

Opis riešenia: Žiaci sa pokúsia o zobrazenie svojho riešenia:

Upravte zdrojový kód triedy `MyWorld` (dvakrát na ňu kliknite) a vytvorte svet s veľkosťou 10x10 buniek. Každá bunka by mala mať veľkosť 75 pixelov.

Commit: [a593cd4a92d0fa0db78275614c3e41a2e96b4e57](https://github.com/uzivatel/uzivatel/commit/a593cd4a92d0fa0db78275614c3e41a2e96b4e57)

6. Konštruktor triedy

Cieľ:

Žiaci zvládnu úvod do konšuktora triedy.

Pojmy na diskusiu:

Konštruktor.

Aktivita:

Učiteľ prezentuje zdrojový kód a predstaví pojem konštruktor.

7. Nastavenia obrázka

Cieľ:

Nastavenie obrázka sveta v projekte Greenfoot.

Pojmy na diskusiu:

Nastavenie obrázka pre triedu `MyWorld`.

Aktivita:

Učiteľ vysvetlí žiakom, že pozadím aplikácie Greenfoot (alebo sveta) môže byť obrázok. Učiteľ vysvetlí žiakom, že pozadie môže byť buď jeden obrázok pokrývajúci celú plochu sveta, alebo obrázok s veľkosťou zodpovedajúcou rozmerom bunky.

Učiteľ ukáže, ako nastaviť obrázok na triede **MyWorld**. Žiaci postupujú podľa pokynov učiteľa a pracujú spoločne krok za krokom.

8. Pripravenie grafiky sveta

Cieľ:

Žiaci sa oboznámia s pozadím triedy World.

Pojmy na diskusiu:

Pozadie triedy **World**.

Aktivita:

Učiteľ zadá žiakom úlohu 1.3. Učiteľ vysvetlí žiakom, čo od nich očakáva. Učiteľ môže stiahnuť záverečný projekt, ktorý už je pripravený vopred, a ukáže im ho. Učiteľ poskytne žiakom odkaz alebo príkaz git na stiahnutie pôvodného projektu, ktorý musia aktualizovať pomocou tejto úlohy (funkcie). Pôvodný projekt si môžu stiahnuť z úložiska git.

Žiaci robia úlohu samostatne alebo v skupine. Učiteľ monitoruje prácu žiakov.

Učiteľ demonštruje riešenie krok za krokom, zatiaľ čo žiaci postupujú podľa pokynov.

Popis riešenia:

Žiaci sa pokúsia o vytvorenie riešenia:

- Nájdite alebo vytvorte vhodný obrázok pre pozadie sveta. Môžete použiť buď pripravené obrázky (vyberte položku Nastaviť obrázok... z kontextového menu triedy MyWorld), alebo vlastný obrázok (skopírujte obrázok do podpričinka images priečinka projektu a vyberte ho rovnakým spôsobom, ako bolo popísané predtým).
- Ako pozadie môžete použiť jediný obrázok, ktorý pokryje celú plochu sveta (vypočítajte potrebnú veľkosť obrázka vzhľadom na veľkosť sveta) alebo menší obrázok, ktorý sa bude opakovane kopírovať (použite štvorcový obrázok s veľkosťou bunky).

Commit: [1184980643db082cfdd6bde9984bceaddf010d49](https://github.com/1184980643db082cfdd6bde9984bceaddf010d49)

3.2. Vytváranie tried a objektov prostredníctvom vývoja hier s Greenfoot

Tabuľka 4: Vytváranie tried a objektov prostredníctvom vývoja hier s Greenfoot

Názov	Vytváranie tried a objektov prostredníctvom vývoja hier s Greenfoot
Výchovno-vzdelávacie ciele a ich opis	Na konci tejto lekcie žiaci pochopia základné pojmy objektu, triedy, vlastnosti a metódy triedy.
Cieľová skupina	Žiaci stredných škôl, ktorí sa zúčastňujú kurzu OOP4Fun. Základné znalosti programovania vrátane objektov, tried, vlastností tried a metód.
Časové trvanie	<ol style="list-style-type: none"> 1. Základné pojmy (25 min) 2. Úloha 1.4 (10 min) 3. Úloha 1.5 (30 min) 4. Rozhranie objektu (5 min) 5. Správa a metóda (15 min) 6. Úloha 1.6 (30 min) 7. Opakovanie teórie (15 min)
Materiály a zdroje	<ul style="list-style-type: none"> • Webová stránka Greenfoot a pokyny na stiahnutie. • Príklady pripravil učiteľ. • Internetové zdroje ako ukážku ďalších príkladov.
Opis	<p>V tejto lekcii žiaci prejdú niekoľkými štruktúrovanými úlohami s cieľom prehĺbiť svoje porozumenie konceptov objektovo orientovaného programovania. Najprv strávia 25 minút vytváraním triedy s názvom Enemy, po ktorej bude nasledovať cieľená 15-minútová úloha na definovanie atribútov a metód v rámci tejto triedy. Následne si vyhradia 30 minút na vytvorenie inštancie objektu triedy Enemy, pričom uplatnia svoje vedomosti o vytváraní a inicializácii objektov. Potom sa v priebehu 5 minút budú venovať koncepcii rozhrania objektu s dôrazom na definovanie toho, aké operácie môže objekt vykonávať. Následne sa žiaci v priebehu 15 minút oboznámia s pojmami správy a metódy, pričom sa naučia, ako objekty komunikujú prostredníctvom volania metód. Ďalších 30 minút bude venovaných praktickej aplikácii, v rámci ktorej budú žiaci posielat</p>

	správy svojej vytvorenej inštancii triedy inštancii Enemy, čím si upevnia svoje chápanie správania objektov. Nakoniec sa v rámci 15-minútovej teoretickej revízie zopakujú kľúčové pojmy, ako sú objekty, triedy, inštancie, vnútorný stav, identita, správy a metódy, čím sa zabezpečí komplexné pochopenie učiva.
<i>Hodnotenie</i>	Gamifikácia predstavuje neformálne hodnotenie, ale zvýši záujem, vnútornú motiváciu a vzdelávacie výstupy celej skupiny.
<i>Šírenie výsledkov</i>	Na šírenie svojich výsledkov učiteľovi a spolužiakom sa použije zvyčajné nastavenie Github/Gitlab a Learning management system (napr. Moodle). Žiaci môžu pokračovať v diskusii na danú tému na fóre, ktoré im je poskytnuté prostredníctvom nástroja na Learning management.

3.2.2. Príručka učiteľa na prípravu vyučovacej hodiny

1. Základný koncept

Cieľ:

Žiaci majú znalosti o triede, jej vlastnostiach a objektoch.

Pojmy na diskusiu:

Triedy, vlastnosti a objekty.

Aktivita:

V úvode hodiny učiteľ so žiakmi zopakuje pojmy, ktoré sa predtým preberali. Prostredníctvom diskusie učiteľ objasní pojmy trieda, vlastnosti tried a objekty.

Učiteľ zadá žiakom písomnú úlohu, v ktorej im dá pokyn, aby v poskytnutom texte identifikovali konkrétne triedy, ich vlastnosti, súvisiace objekty a príslušné hodnoty.

Učiteľ vyberie žiaka z triedy, ktorý predstaví svoje riešenie. Počas tejto aktivity sa ostatní žiaci zúčastňujú tým, že vyjadrujú svoje názory.

Vyučujúci krok za krokom popíše, ako vytvoriť triedu v prostredí Greenfoot. Žiaci postupujú podľa pokynov učiteľa a spolupracujú krok za krokom.

2. Úloha 1.4

Cieľ:

Žiak sa vie zapojiť do práce na projekte.

Pojmy na diskusiu:

Vytvorenie triedy.

Aktivita:

Učiteľ zadá žiakom úlohy, Úloha 1.4.

Učiteľ vysvetlí žiakom, čo od nich očakáva. Učiteľ môže stiahnuť záverečný projekt, ktorý už pripravil vopred, a ukázať im ho. Učiteľ poskytne žiakom odkaz alebo príkaz git na stiahnutie pôvodného projektu, ktorý musia aktualizovať pomocou tejto úlohy (funkcie). Pôvodný projekt pre túto úlohu si môžete stiahnuť z repozitára git.

Žiaci úlohu riešia samostatne alebo v skupine. Učiteľ monitoruje prácu žiakov. učiteľ demonštruje riešenie krok za krokom, pričom žiaci postupujú podľa pokynov. Opis riešenia: Žiaci sa pokúsia nájsť riešenie, ktoré by im pomohlo pri riešení úlohy:

Vytvorte nepriateľa. Nepriateľ bude pochodovať smerom k hráčovej guli, aby ju poškodil a nakoniec zničil. Vytvorte novú podtriedu triedy Actor (vyberte položku Actor kliknutím pravým tlačidlom myši a z kontextového menu triedy Actor vyberte položku „Newsubclass“...). Učiteľ by mal vysvetliť konvenciu, že trieda by mala začínať veľkými písmenami, a celú konvenciu názvov v Jave.

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

3. Úloha 1.5

Cieľ:

Žiak porozumie stavu objektu. Žiak bude vedieť vytvoriť inštanciu triedy v Greenfoot.

Pojmy na diskusiu:

Stav objektu, inštancia.

Aktivita:

Učiteľ vysvetlí pojem stav objektu na jednoduchom príklade. Napríklad učiteľ je umiestnený v triede v určitej vzdialenosti od vchodových dverí. Táto vzdialenosť určuje ich polohu, pričom kroky vpred alebo vzad menia ich blízkosť k dverám. Preto možno polohu učiteľa vzhľadom na vchodové dvere kvantifikovať pomocou premennej, ktorá sa mení v čase. Učiteľova poloha je teda definovaná vzdialenosťou od vchodových dverí, čo ilustruje, ako je stav objektu - v tomto prípade učiteľa - charakterizovaný hodnotou jeho atribútu (vzdialenosti) v danom okamihu.

Učiteľ opíše úlohu 1.5, ako vytvoriť inštanciu triedy Enemy. Učiteľ otvorí poslednú verziu projektu, zatiaľ čo žiaci postupujú podľa pokynov učiteľa a pracujú spoločne krok za krokom.

Učiteľ vytvorí inštanciu triedy Enemy (vyberie položku pravým tlačidlom myši new Enemy () z kontextového menu triedy Enemy, umiestni inštanciu do sveta kliknutím ľavým tlačidlom myši na požadovanú pozíciu). Preskúmajte jej vnútorný stav (vyberte prvok umiestnený do sveta kliknutím pravým tlačidlom myši a z kontextového menu vytvorenej inštancie vyberte položku Inspect).

Učiteľ požiada žiakov, aby vytvorili novú inštanciu a umiestnili ju na inú pozíciu a porovnali vnútorné stavy dvoch vytvorených inštancií.

4. Rozhranie objektu

Cieľ:

Žiaci sa oboznámia s rozhraním ako súborom akcií, ktoré môžeme vykonať na nejakom objekte.

Pojmy na diskusiu:

Rozhranie objektu.

Aktivita:

Učiteľ oboznámi žiakov s pojmom rozhranie na jednoduchých príkladoch. Napríklad, ak pozorujeme človeka a činnosti, ktoré vykonáva počas dňa (ako je prebudenie, raňajky, cesta do práce) bez toho, aby sme sa zaoberali konkrétnymi detailmi - ako sa budí (či budíkom, telefónom alebo rodičom), čo a kde raňajkuje alebo akým spôsobom sa presúva do práce - súbor týchto činností možno prirovnať k rozhraniu. Rozhranie definuje, aké činnosti môžu objekty určitej triedy vykonávať, ale nešpecifikuje, ako sa tieto činnosti vykonávajú. Nedefinujeme, čo si dá na raňajky, ako išiel do práce (pešo, autom alebo autobusom) alebo ako sa zobudil (či ho niekto zavolať, alebo ho zobudili hodiny).

5. Správa a metóda

Cieľ:

Žiaci sa oboznámia s pojmom metóda.

Pojmy na diskusiu:

Metóda objektu.

Aktivita:

Učiteľ predstaví pojem metóda. Na vysvetlenie pojmu metóda učiteľ spojí pojmy vlastnosti (charakteristiky, atribúty) a meniace sa hodnoty týchto vlastností.

Príklad: V prípade, že sa na základe údajov o vlastnostiach (napr. Ak uvažujeme triedu **Osoba** a vlastnosť vek, jej hodnota sa každý rok zvyšuje o jednotku, a to dôsledne v ten istý deň. Na druhej strane, ak sa pozrieme na vlastnosti ako výška a hmotnosť, tieto vlastnosti sa často menia - ľudia rastú a ich hmotnosť kolíše.

Príklad: Ak pozorujeme pohyb človeka z bodu A do bodu B a tento pohyb opíšeme v krokoch - napríklad vykročenie dopredu, otočenie doľava o 45 stupňov, urobenie 8 krokov dopredu, otočenie doprava o 30 stupňov a urobenie ďalších 5 krokov dopredu - tieto jednotlivé činnosti môžeme zoskupiť do tzv. metódy.

Učiteľ ukáže žiakom v Greenfoot, ako sa môžu pozrieť na metódy, ktoré má trieda a ktoré sa dajú zavolať na konkrétnom objekte.

Učiteľ ukáže metódy definované v triede **Actor**. Učiteľ ukáže, ako zavolať metódy objektu.

6. Úloha 1.6

Cieľ:

Žiaci sa oboznámia, ako zavolať metódu na objekte inštancie.

Pojmy na diskusiu:

Volanie metód.

Aktivita:

Učiteľ popíše úlohu 1.6. Učiteľ otvorí poslednú verziu projektu, zatiaľ čo žiaci postupujú podľa pokynov učiteľa a pracujú spoločne krok za krokom.

Učiteľ odošle správu inštancii triedy **Enemy**, aby sa presunula na pozíciu [12, 6] a bola otočená smerom nadol (kliknutím pravým tlačidlom myši na objekt vo **World**, výberom *inherited from Actor* a potom výberom metódy **setLocation(int, int)**). Popíšte žiakom, čo sa stane s inštanciou. Ako bol ovplyvnený vnútorný stav príslušnej inštancie?

Učiteľ demonštruje žiakom, ako sa definujú metódy. Vytvorí metódu **setPosition(int x, int y)** na nastavenie inštancie triedy **Actor** na konkrétne súradnice. Učiteľ zdôrazní, že táto metóda je ekvivalentná metóde **setLocation(int x, int y)**, a zdôrazní, že je dôležité pred definovaním novej metódy skontrolovať, či už metóda neexistuje, aby sa predišlo duplicitě. Zdôrazňuje, že názvy metód by mali jasne naznačovať ich účel a funkčnosť, čo umožňuje okamžité pochopenie už len z názvu. Učiteľ tiež zdôrazňuje, že názvy metód by mali byť stručné, a uvádza príklady dobre definovaných, zle definovaných a nesprávne definovaných metód. Metódy, ktoré môže používateľ vykonať (zavolať), sú viditeľné po kliknutí pravým tlačidlom myši na objekt.

Učiteľ popíše, ako definovať metódu. Žiaci postupujú podľa pokynov učiteľa a pracujú spoločne krok za krokom.

Učiteľ požiada žiakov, aby definovali metódu, ktorou sa aktér znižuje (znižuje súradnicu *x*), a metódu, ktorou sa zvyšuje (zvyšuje súradnicu *y*). Učiteľ sleduje žiakov, ako pracujú na úlohe, a ak je to potrebné, dáva pokyny po jednom.

7. Opakovanie teórie

Cieľ:

Žiaci rozumejú pojmom, ktoré sa preberali.

Pojmy na diskusiu:

Objekt, trieda, inštancia, vnútorný stav, identita, správa, metóda.

Aktivita:

Učiteľ zhrnie koncept, ktorý sa preberal na vyučovaní.

4. Algoritmus

V rámci tematického celku Algoritmus boli vytvorené dva metodické materiály.

4.1. Úvod do algoritmov v prostredí Greenfoot

Tabuľka 5: Úvod do algoritmov a algoritmického myslenia

Názov	Úvod do algoritmov a algoritmického myslenia
Výchovno-vzdelávacie ciele a ich opis	Na konci lekcie by mali žiaci dôkladne porozumieť algoritmom a algoritmickému mysleniu, mali by vedieť navrhovať a implementovať základné algoritmy a mali by byť schopní efektívne používať algoritmické koncepty na riešenie rôznych problémov.
Cieľová skupina	Žiaci stredných škôl, ktorí sa zúčastňujú kurzu OOP4Fun. Žiaci by mali mať základné znalosti programovania vrátane premenných, funkcií, pojmov iterácie a vetvenia, zručnosti v logickom uvažovaní a v riešení problémov. Žiaci by mali byť oboznámení s programom Greenfoot.
Časové trvanie	<ol style="list-style-type: none">1. Úvod do základných algoritmov ako postupnosť krokov (15 min)2. Úloha 2.1 - Napísanie jednoduchého algoritmu (20 min)3. Algoritmus a jeho vlastnosti (15 min)4. Úloha 2.2 - Napísanie všeobecnejšieho algoritmu (25 min)5. Algoritmizácia (15 min)
Materiály a zdroje	<ul style="list-style-type: none">• Učebnica z projektu OOP4Fun.• Zdroje z projektu OOP4Fun.• Zdrojový kód projektu z Github/Gitlab.• Internetové zdroje.
Opis	<p>V tomto metodickom materiáli sa žiaci stredných škôl ponoria do oblasti algoritmov a algoritmického myslenia. Hodina sa začína 15-minútovým úvodom zameraným na oboznámenie žiakov so základnými algoritmickými pojmami a zdôraznenie ich významu pri riešení problémov.</p> <p>Po úvode sa žiaci zapoja do 20-minútovej úlohy, v ktorej majú za úlohu napísať jednoduchý algoritmus na riešenie konkrétneho problému. Táto praktická aktivita umožňuje žiakom aplikovať pojmy, ktoré boli predstavené predtým, a zdokonaľiť svoje algoritmické zručnosti.</p>

	<p>Následne bude 15-minútový úsek venovaný diskusii o algoritmoch a ich vlastnostiach. Medzi preberané témy bude patriť správnosť, efektívnosť a škálovanie, pričom sa zdôrazní dôležitosť jasných a presných inštrukcií pri návrhu algoritmov.</p> <p>Na základe svojich poznatkov stravia žiaci ďalších 25 minút vytváraním všeobecnejšieho algoritmu pre mierne zložitý problém. Táto úloha je pre žiakov výzvou na abstraktné a kritické myslenie a na uplatňovanie algoritmických princípov pri riešení reálnych scenárov.</p> <p>V záverečnom 15-minútovom úseku sa žiaci budú venovať algoritmickej analýze a zdokonaľovaniu svojich algoritmov. Tento proces zahŕňa identifikáciu potenciálnych zlepšení, optimalizáciu účinnosti a zabezpečenie robustnosti algoritmov.</p> <p>Počas celého stretnutia budú žiaci pracovať individuálne alebo v malých skupinách, čo podporí spoluprácu a vzájomné učenie. Aktívnou účasťou na písaní a analýze algoritmov si žiaci rozvinú schopnosti kritického a algoritmického myslenia.</p> <p>Na záver stretnutia budú žiaci hlbšie rozumieť algoritmom a algoritmickému mysleniu, čo ich vybaví základnými zručnosťami pre systematický a efektívny prístup ku komplexným problémom.</p>
<i>Hodnotenie</i>	Gamifikácia predstavuje neformálne hodnotenie, ale zvýši záujem, vnútornú motiváciu a vzdelávacie výstupy celej skupiny.
<i>Šírenie výsledkov</i>	Na šírenie svojich výsledkov učiteľovi a spolužiakom sa použije zvyčajné nastavenie Github/Gitlab a Learning management system (napr. Moodle). Žiaci môžu pokračovať v diskusii na danú tému na fóre, ktoré im je poskytnuté prostredníctvom nástroja na Learning management.

4.1.2. Príručka učiteľa na prípravu vyučovacej hodiny

1. Úvod do základných algoritmov ako postupnosť krokov

Cieľ:

Žiaci sa oboznámia s úvodom do algoritmov.

Pojmy na diskusiu:

Základy algoritmu.

Aktivita:

Učiteľ predstaví pojem algoritmus na príkladoch z reálneho života. Učiteľ sa napríklad pýta žiakov, čo robia ráno od prebudenia až po príchod do školy. Potom sa učiteľ spýta žiakov, či vedia, ako sa robí pizza alebo teplý sendvič, alebo či niekto z nich pozná recept na prípravu jedla alebo koláča.

Učiteľ spojí proces prípravy jedla alebo koláča s tvorbou programu a zdôrazní, že tak ako existuje recept na prípravu jedla, existuje aj „recept“ na tvorbu programu nazývaného algoritmus.

Učiteľ dospeje k záveru, že algoritmus je súbor krokov, ktoré definujú, ako sa program vykonáva.

Učiteľ ďalej vysvetlí, že tento súbor krokov sa nemusí vždy vykonávať postupne, jeden po druhom, že v algoritme sú niektoré kroky, ktoré sa môžu vykonávať v závislosti od nejakej podmienky. Učiteľ požiada žiakov, aby uviedli príklad takéhoto prípadu (napríklad ak píšeme algoritmus na prípravu teplého sendviča a nemáme nejakú ingredienciu, napríklad šunku, ale máme podobnú ingredienciu, môžeme ju nahradiť, alebo ísť do obchodu a chýbajúcu ingredienciu zohnať).

Učiteľ vysvetlí žiakom, že niektoré kroky v algoritme sa môžu opakovať niekoľkokrát. Učiteľ požiada žiakov, aby uviedli príklad algoritmu, v ktorom sa kroky opakujú niekoľkokrát.

2. Úloha 2.1 - Napísanie jednoduchého algoritmu

Cieľ:

Žiaci sa naučia písať jednoduchý algoritmus.

Pojmy na diskusiu:

Algoritmus.

Aktivita:

Učiteľ zadá žiakom úlohu 2.1, aby na papier napísali postup, ktorým opíšu, ako chodec prechádza cez ulicu.

Na začiatku učiteľ nedáva žiakom ďalšie pokyny, ale sleduje, ako rozmýšľajú a pracujú. Ak niekto položí otázku, ktorá je dôležitá pre opis postupu prechodu chodca cez ulicu, pochváli ho a zdôrazní, prečo je táto informácia dôležitá.

Po určitom čase sa učiteľ spýta žiakov, či venovali pozornosť tomu, kde chodec prechádza cez ulicu, či je to vyznačené miesto na prechod alebo nie. Taktiež sa učiteľ spýta žiakov, či sa zamysleli nad tým, či je na priechode pre chodcov semafor alebo nie.

Na záver učiteľ vyberie niekoľko žiakov, ktorí by mali prečítať svoje pokyny na prechod cez ulicu.

3. Algoritmus a jeho vlastnosti

Cieľ:

Žiaci sa oboznámia s vlastnosťami algoritmu

Pojmy na diskusiu:

Vlastnosti algoritmu.

Aktivita:

Učiteľ ďalej vysvetlí žiakom vlastnosti algoritmu. Učiteľ vysvetlí žiakom, že algoritmy možno znázorniť aj graficky, a uvedie príklad algoritmov, ktoré sú znázornené graficky.

4. [Úloha 2.2 - Napísanie všeobecnejšieho algoritmu](#)

Cieľ:

Žiak dokáže napísať všeobecnejší algoritmus

Pojmy na diskusiu:

Zápis algoritmu.

Aktivita:

Vytvorte všeobecný algoritmus prípravy horúceho nápoja. Premyslite si, aké musia byť vstupy takéhoto algoritmu, aby bol všeobecný.

5. [Algoritmizácia](#)

Cieľ:

Žiaci vedia programovať viac príkladov a sú zapojení do definovania vlastných

Pojmy na diskusiu:

Algoritmus.

Aktivita:

Učiteľ vysvetlí žiakom, že aj v matematike existujú určité algoritmy, ktoré používame pri riešení problémov. Učiteľ sa spýta žiakov, či môžu uviesť nejaký príklad.

Príklad, ktorý učiteľ žiakom vysvetlí, je príklad výpočtu hodnoty aritmetického výrazu, ktorý má niekoľko matematických operácií, kde je potrebné dodržiavať pravidlá priority vykonávania matematických operácií.

Učiteľ uvedie aj ďalšie príklady, napríklad montáž nového nábytku, ktorý bol zakúpený a prišiel s návodom, ktorý opisuje, ako tento nábytok zostaviť. Príkladom môžu byť pokyny, ktoré dostávame prostredníctvom zariadenia GPS, keď sa chceme dostať z bodu A do bodu B prostredníctvom navigácie.

Učiteľ požiadajú žiakov, aby na papier napísali vlastný algoritmus, po ktorom niektorí z nich prezentujú výsledok.

4.2. Greenfoot dobrodružstvá: Volanie metód v Java, práca s dokumentáciou a riadenie aplikácie

Tabuľka 6: Greenfoot dobrodružstvá: Volanie metód v Java, práca s dokumentáciou a riadenie aplikácie

Názov	Greenfoot dobrodružstvá: Volanie metód v Java, práca s dokumentáciou a riadenie aplikácie
Výchovno-vzdelávacie ciele a ich opis	Na konci lekcie by mali žiaci dobre porozumieť volaniu metód jazyka Java, najmä so zameraním na metódy „act()“ a „move()“ v prostredí Greenfoot. Mali by ovládať efektívne používanie kľúčového slova „this“ ako referenciu na aktuálne objekty v kontexte triedy. Okrem toho by žiaci mali preukázať schopnosť volať metódy v rámci triedy, pochopiť syntax a parametre potrebné na volanie metód. Mali by tiež preukázať zručnosť pri používaní techník volania metód na efektívne riešenie úloh interaktívneho vývoja hier. Okrem toho by žiaci mali pochopiť dôležitosť dokumentácie programu a mali by byť schopní efektívne dokumentovať program v jazyku Java a zabezpečiť jeho prehľadnosť a čitateľnosť. Nakoniec by mali ovládať techniky ovládania aplikácií v rámci Greenfoot, ktoré umožňujú presnú manipuláciu a interakciu s hernými prvkami na vytvorenie pútavej a funkčnej hernej mechaniky.
Cieľová skupina	Žiaci stredných škôl na kurze OOP4Fun. Základné znalosti programovania vrátane konceptov iterácie a vetvenia. Žiaci by mali byť oboznámení s programom Greenfoot.
Časové trvanie	<ol style="list-style-type: none"> 1. Vysvetlenie metódy act () (10 min) 2. Vysvetlenie metódy move() (20 min) 3. Predstavenie kľúčového slova this (5 min) 4. Úloha 2.3 - Zavolanie metódy (10 min) 5. Vysvetlenie automatického dopĺňania (5 min) 6. Význam dokumentácie programu (15 min) 7. Úloha 2.4 - Pridanie dokumentácie (5 min) 8. Úloha 2.5 - Pridanie ďalšej dokumentácie (5 min) 9. Úloha 2.6 - Prečítanie si dokumentácie z projektu Bomberman (10 min) 10. Úloha 2.7 - Preskúmanie ovládacích prvkov aplikácie z projektu Bomberman (20 min) 11. Diskusia: Algoritmus, vlastnosti, algoritmizácia, tlačidlá Greenfoot (5 min)

<p><i>Materiály a zdroje</i></p>	<ul style="list-style-type: none"> • Učebnica z projektu OOP4Fun. • Zdroje z projektu OOP4Fun. • Zdrojový kód projektu z Github/Gitlab. Internetové zdroje.
<p><i>Opis</i></p>	<p>V tomto 105-minútovom metodickom materiáli sa žiaci stredných škôl vydajú na cestu k zvládnutiu volania metód jazyka Java, dokumentácie programu a riadenia aplikácie v prostredí Greenfoot.</p> <p>Hodina sa začína 15-minútovým skúmaním metódy „act()“, po ktorom nasleduje 10 minútový úvod do metódy „move()“, ktorá je základnou súčasťou programovania v Greenfoot.</p> <p>Žiaci sa potom 5 minút budú venovať významu kľúčového slova „this“ pri odkazovaní na aktuálne objekty v kontexte triedy.</p> <p>Potom na žiakov čaká 10-minútová úloha, ktorá ich vyzve, aby si precvičili volanie metód v rámci triedy, pričom použijú syntax a parametre potrebné na volanie metód.</p> <p>Nasleduje 5-minútové vysvetlenie funkcií automatického dokončovania v prostredí Greenfoot, ktoré sa zameriava na zvýšenie efektívnosti programovania.</p> <p>Počas ďalších 15 minút žiaci pochopia dôležitosť dokumentácie programu, pričom pochopia, ako jasná a stručná dokumentácia zvyšuje čitateľnosť a udržiavateľnosť programu. V 5-minútovej úlohe žiaci doplnia dokumentáciu k svojmu kódu, čím zabezpečia jasnosť a zrozumiteľnosť pre seba aj ostatných. V nadväznosti na túto úlohu stravia žiaci ďalších 5 minút pridávaním podrobnejšej dokumentácie k svojmu kódu. v 10-minútovej úlohe budú žiaci skúmať a čítať dokumentáciu pridanú ich rovesníkmi, čím získajú prehľad o rôznych štýloch a prístupoch k programovaniu.</p> <p>Nakoniec žiaci stravia 20 minút skúmaním techník ovládania aplikácií v rámci Greenfoot, ktoré umožňujú presnú manipuláciu a interakciu s hernými prvkami na vytvorenie pútavej a funkčnej hernej mechaniky.</p> <p>Počas celého stretnutia budú žiaci pracovať individuálne alebo v malých skupinách, čo podporí spoluprácu a vzájomné učenie. Aktívnou účasťou na písaní a analýze kódu si žiaci rozvinú schopnosti kritického myslenia a riešenia algoritmických problémov.</p> <p>Na konci stretnutia žiaci získajú hlbšie pochopenie volania metód jazyka Java, dokumentácie kódu a riadenia aplikácií v Greenfoot, čím budú vybavení základnými zručnosťami pre vývoj hier aj mimo neho.</p>

<i>Hodnotenie</i>	Gamifikácia predstavuje neformálne hodnotenie, ale zvýši záujem, vnútornú motiváciu a vzdelávacie výstupy celej skupiny.
<i>Šírenie výsledkov</i>	Na šírenie svojich výsledkov učiteľovi a spolužiakom sa použije zvyčajné nastavenie Github/Gitlab a Learning management system (napr. Moodle). Žiaci môžu pokračovať v diskusii na danú tému na fóre, ktoré im je poskytnuté prostredníctvom nástroja na Learning management.

4.2.2. Príručka učiteľa na prípravu vyučovacej hodiny

1. Vysvetlenie metódy `act()`

Cieľ:

Žiaci porozumejú použitiu metódy `act()` a volanie metódy z inej metódy.

Pojmy na diskusiu:

Metóda `act()`.

Aktivita:

Učiteľ otvorí poslednú verziu projektu **TowerDefense**. Učiteľ umiestni inštancie triedy **Enemy** na pozíciu 0,0. Učiteľ sa spýta žiakov, čo sa stane, keď zavoláme metódu `act()` na inštanciu triedy **Enemy**. Je to očakávané správanie?

Učiteľ požiada žiaka, aby mu pomohol splniť úlohu, posunúť objekt triedy **Enemy** o dve bunky v aktuálnom smere, keď zavoláme metódu `act()`.

Učiteľ ukáže žiakom, ako túto úlohu vykonať (pridá volanie metódy `move(int)` do vnútra metódy `act()`), pričom dáva pokyny a žiaci hovoria svoje nápady.

2. Vysvetlenie metódy `move()`

Cieľ:

Vysvetlite metódu `move()` a spôsob presunu dopredu a dozadu.

Pojmy na diskusiu:

Metódy pre pohyb.

Aktivita:

Učiteľ vysvetlí metódu `move(int)`. Učiteľ umiestni inštancie triedy **Enemy** na rôzne pozície a zavolá metódu odovzdávajúcu kladné hodnoty, napríklad 1 alebo 3. Učiteľ sa spýta žiakov, čo si myslia, že sa stane, ak zavolajú metódu `move()` a zadajú záporné hodnoty, napríklad -1.

Učiteľ zadá žiakom úlohu napísať metódu **backward()**, ktorá posunie objekt o 1 krok dozadu. Je možné posunúť objekt o 2 kroky dozadu a o X krokov? Čo musíme urobiť?

3. [Predstavenie kľúčového slova this](#)

Cieľ:

Žiak porozumie zavedeniu kľúčového slova **this**.

Pojmy na diskusiu:

Kľúčové slovo **this**.

Aktivita:

Učiteľ zavádza kľúčové slovo **this**.

4. [Úloha 2.3 - Zavolanie metódy](#)

Cieľ:

Žiaci budú vedieť, ako napísať metódu na vertikálny pohyb objektu, ktorá mení hodnotu osi y.

Pojmy na diskusiu:

Volanie metódy.

Aktivita:

Učiteľ sa spýta žiakov, ako sa má pohybovať objektom vertikálne, hore alebo dole? Učiteľ požiada žiaka, aby našiel vhodnú metódu pre pohyb hore/dole kliknutím pravým tlačidlom myši na objekt a výberom možnosti *Inherited from Actor*. Žiak by mal rozpoznať nasledujúce metódy: **turn**, **setRotation**, **setLocation**, **getLocation**, **getRotation**. učiteľ zadá žiakom úlohu napísať metódu **up()** a **down()**. Učiteľ sleduje prácu žiakov a na záver spolu s nimi vysvetlí, ako tieto metódy implementovať. Metódy **up()** a **down()** menia hodnotu súradnice y, zvyšujú ju, resp. znižujú. Tu môže učiteľ predstaviť parametre metódy, ale bez podrobností.

5. [Vysvetlenie automatického dopĺňania](#)

Cieľ:

Žiaci sa oboznámia s automatickým dopĺňaním v prostredí Geenfoot.

Pojmy na diskusiu:

Automatické dopĺňanie - autocomplete (CTRL+SPACE).

Aktivita:

Učiteľ popíše žiakom, ako nájsť metódu na objekte, ktorú môžu zavolať, ak zabudli názov alebo ak práve začínajú programovať a chcú „požiadať“ prostredie Greenfoot, aby im pomohlo rýchlejšie dokončiť písanie kódu.

6. Význam dokumentácie programu

Cieľ:

Žiaci sa oboznámia s komentármi a dokumentáciou.

Pojmy na diskusiu:

Komentáre a dokumentácia, preskúmanie okna dokumentácie.

Aktivita:

Učiteľ oboznámi žiakov s príkazom v programe, ktorý nie je súčasťou vykonávania (komentáre, dokumentačné komentáre).

Učiteľ musí zdôrazniť rozdiely medzi komentármi a dokumentáciou (ako špeciálnym typom komentárov).

Učiteľ predstaví žiakom zdrojový kód triedy **Enemy** a následne vygeneruje html dokument, ktorý popisuje dokumentáciu triedy **Enemy**. Učiteľ tu popíše, že existuje určité pravidlo, ktoré musíme dodržiavať pri písaní dokumentácie pre našu triedu alebo metódy.

Učiteľ zadá žiakom úlohu, aby preskúmali, ako písať dokumentáciu k triedam a metódam jazyka Java.

7. Úloha 2.4 - Pridanie dokumentácie

Cieľ:

Žiak sa zoznámia s dokumentačnými komentármi k metóde.

Pojmy na diskusiu:

Dokumentačný komentár k metóde.

Aktivita:

Učiteľ zadá žiakom úlohu pridať komentár k dokumentácii metódy `act()`.

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

8. Úloha 2.5 - Pridanie ďalšej dokumentácie

Cieľ:

Žiak sa zoznámia s dokumentačnými komentármi k triede.

Pojmy na diskusiu:

Dokumentačný komentár k triede.

Aktivita:

Učiteľ zadá žiakom úlohu pridať dokumentačný komentár k metóde `act()`.

Upravte komentár k dokumentácii triedy **Enemy**. Pridajte verziu triedy a jej autora a pozrite si zmeny na vygenerovanej webovej stránke v jazyku HTML.

Commit: [1a7a9f83c5271a7c0dfa46ce3b1ee65682b0c5e5](#)

9. Úloha 2.6 - Prečítanie si dokumentácie z projektu Bomberman

Cieľ:

Žiaci sa naučia skúmať dokumentáciu triedy s cieľom pochopiť metódy.

Pojmy na diskusiu:

Preskúvanie dokumentácie.

Aktivita:

Učiteľ zadá žiakom úlohu preskúmať okno s dokumentáciou a prečítať si dokumentáciu k triedam **Actor** a **World**. Učiteľ zdôrazní dôležitosť čítania dokumentácie s cieľom nájsť metódy, ktoré môžu byť užitočné.

10. Úloha 2.7 - Preskúvanie ovládacích prvkov aplikácie z projektu Bomberman

Cieľ:

Žiaci sa oboznámia s tlačidlami Greenfoot.

Pojmy na diskusiu:

Greenfoot tlačidlá.

Aktivita:

Učiteľ pokračuje v práci na poslednej verzii projektu. Učiteľ požiadajú žiakov, aby pridali dve inštancie objektu triedy **Enemy** a zavolali metódu **act()** na každej inštancii.

Učiteľ popíše tlačidlo **Act**. Učiteľ klikne na tlačidlo **Act** umiestnené v hlavnom okne. Učiteľ im položí otázku, aby vysvetlili, čo sa stalo.

Učiteľ tiež požiadajú žiaka, aby klikol na tlačidlo **Run** a vysvetlili, čo sa stalo.

Učiteľ požiadajú žiakov, aby klikli na tlačidlo **Reset** a vysvetlili, čo sa stalo. Potom učiteľ vysvetlí, čo je potrebné urobiť, aby sa pri každom kliknutí na tlačidlo **Reset** na tabuli objavili dva objekty triedy **Enemy** na pozíciách (0,3) a (3,3).

Učiteľ by mal žiakom vysvetliť, že ak chce, aby sa tieto objekty objavili na tabuli vždy po kliknutí na tlačidlo **Reset**, je potrebné zmeniť konštruktor triedy **World** tak, aby sa tieto objekty vytvorili v konštruktoze a umiestnili na požadované pozície.

11. Diskusia: Algoritmus, vlastnosti, algoritmizácia, tlačidlá Greenfoot

Cieľ:

Žiaci dokážu zhrnúť to, čo sa naučili v lekcii.

Pojmy na diskusiu:

Metódy pohybu, tlačidlá Greenfoot, dokumentácia.

Aktivita:

Učiteľ zhrnie učivo. Učiteľ tu môže zdôrazniť dôležitosť zdrojového kódu, pravidiel pre pomenovanie metód a tried.

5. Vetvenie

V rámci tematického celku Vetvenie boli vytvorené dva metodické materiály.

5.1. Skúmanie vetvenia pri vývoji hier s Greenfoot - Neúplné vetvenie kódu

Tabuľka 7: Skúmanie vetvenia pri vývoji hier s Greenfoot - Neúplné vetvenie kódu

<i>Názov</i>	Skúmanie vetvenia pri vývoji hier s Greenfoot - Neúplné vetvenie kódu.
<i>Výchovno-vzdelávacie ciele a ich opis</i>	Téma zahŕňa neúplné vetvenie (viacnásobné vetvenie je zámerne vynechané). Uvádzajú sa základy vnímania sveta aktorm. Žiaci budú schopní písať kód pomocou podmienok. Po ukončení tejto témy sa žiaci naučia používať jednoduché príkazy if-else a naučia sa robiť rozhodnutia v kóde, ktoré kontrolujú správanie ich hry. Získajú základné znalosti programovacieho jazyka Java, naučia sa potrebnú syntax a analyzovať a pochopiť kód, čo im pomôže pochopiť, prečo sa ich hra správa určitým spôsobom a ako odstrániť problémy. Žiaci budú môcť vytvoriť vlastné herné projekty s využitím toho, čo sa naučili o objektovo orientovanom programovaní.
<i>Cieľová skupina</i>	Žiaci stredných škôl na kurze OOP4Fun. Základné znalosti programovania vrátane premenných a dátových typov. Žiaci by mali byť oboznámení s programom Greenfoot.
<i>Časové trvanie</i>	<ol style="list-style-type: none">1. Úvod (5 minutes)2. Vysvetlenie kódu (15 minutes)3. Neúplné vetvenie (10 minutes)4. Pozorovanie stavu hráča (10 min)5. Úloha 3.2 - Pridanie detekcie hraníc sveta (10 min)
<i>Materiály a zdroje</i>	<ul style="list-style-type: none">• Učebnica z projektu OOP4Fun.• Zdroje z projektu OOP4Fun.• Zdrojový kód projektu z Github/Gitlab.• Internetové zdroje.
<i>Popis</i>	Hodina sa začína 15-minútovým vysvetlením kódu metódy <code>turn()</code> , ktorá pripravuje pôdu pre pochopenie neúplného vetvenia kódu. Túto diskusiu vedie učiteľ a jej cieľom je priviesť žiakov k spoločnému

	<p>pochopeniu úlohy tejto metódy vo svete aktorov.</p> <p>Po nej nasleduje 10-minútový blok venovaný osvojeniu si základných pojmov súvisiacich s neúplným vetvením. Táto fáza osvojovania je pre žiakov kľúčová, aby si osvojili základné princípy predtým, ako sa pustia do písania kódu.</p> <p>Následne sa žiaci venujú 10-minútovej úlohe skúmania, v ktorej pozorujú stav hráča v rámci svojho kódu, čo podporuje hlbšie pochopenie toku programu na základe zdrojového kódu.</p> <p>Následná 10-minútová produkčná fáza zahŕňa pridanie detekcie okrajov sveta do ich projektu, pridanie správania a nastavenie hry.</p>
<i>Hodnotenie</i>	<p>Táto aktivita umožní učiteľom poskytovať formatívnu spätnú väzbu na základe diskusií a monitorovania žiakov v obrátenej triede.</p> <p>Vzájomné hodnotenie sa bude vykonávať online ako súčasť domácej úlohy. Žiakom to pripomenie dôležité aspekty úlohy, prinúti ich kriticky hodnotiť prácu iných žiakov, poskytne im to prehľad o dobrých alebo nie veľmi dobrých riešeniach ich rovesníkov atď. a zvýši to celkové dosiahnutie výsledkov vzdelávania.</p> <p>Pri práci v tímovom projekte, na ktorom žiaci pracujú, sa budú využívať aj tieto výsledky vzdelávania a poznatky.</p>
<i>Šírenie výsledkov</i>	<p>Na šírenie svojich výsledkov učiteľovi a spolužiakom sa použije zvyčajné nastavenie Github/Gitlab a Learning management system (napr. Moodle). Žiaci môžu pokračovať v diskusii na danú tému na fóre, ktoré im je poskytnuté prostredníctvom nástroja na Learning management.</p>

5.1.2. Príručka učiteľa na prípravu vyučovacej hodiny

1. Úvod

Cieľ:

Žiaci s učiteľom prediskutujú pojmy, ktoré sa preberali na predchádzajúcej hodine.

Koncepty na diskusiu:

Algoritmus, dokumentácia kódu.

Aktivita:

V úvodnej časti hodiny si učiteľ so žiakmi zopakuje pojmy osvojené z predchádzajúcej hodiny. Potom učiteľ predstaví novú látku, ktorá sa bude na dnešnej hodine preberať. Na ilustráciu dnešnej hodiny učiteľ uvedie príklad algoritmu s jednoduchým vetvením.

Príkladom môže byť algoritmus prechodu cez ulicu na priechode pre chodcov, kde nie je semafor. Chodec neprejde ulicu hneď, ale najprv sa uistí, že zľava ani sprava neprichádzajú žiadne vozidlá. Ak tam nie sú žiadne vozidlá, chodec prejde cez ulicu.

2. Vysvetlenie kódu

Cieľ:

Žiak porozumie vlastnostiam triedy.

Koncepty na diskusiu:

Vlastnosti triedy.

Aktivita:

Učiteľ stiahne najnovšiu verziu projektu:

- z platformy moodle alebo
- z git repozitára.

Učiteľ vytvorí inštanciu triedy **Enemy** a umiestni ju niekde do sveta. Vysvetlí niektoré metódy triedy **Actor**:

- `move(int)`
- `turn(int)`
- `setRotation()`

Pri vysvetľovaní metód učiteľ tiež ukazuje, ako sa menia niektoré vlastnosti triedy (napríklad poloha objektu vo svete, t. j. hodnoty atribútov `x` a `y`). Učiteľ diskutuje so žiakmi o tom, ako doplniť metódu `act()` tak, aby sa pri každom volaní metódy `act()` inštancia triedy **Enemy** posunula o dva kroky dopredu.

3. Neúplné vetvenie

Cieľ:

Žiak dokáže vytvoriť metódu `showText()`.

Koncepty na diskusiu:

Zobrazenie textu v projekte, vetvenie, neúplné vetvenie.

Aktivita:

Učiteľ pokračuje v práci na projekte. Učiteľ umiestni do sveta inštanciu triedy **Enemy**. Učiteľ vysvetlí žiakom, ako môžu skontrolovať, či sa objekt nachádza v hornej polovici sveta, a zobrazí správu „*Nájdený*“.

Učiteľ vysvetlí žiakom metódu `showText()`, ktorá sa používa na zobrazenie textu.

4. Pozorovanie stavu hráča

Cieľ:

Žiak dokáže analyzovať vnútorný stav inštancie.

Koncepty na diskusiu:

Vnútorný stav inštancie, atribúty `x`, `y`, rotácia.

Aktivita:

Učiteľ vytvorí inštanciu triedy **Enemy** a umiestni ju do stredu sveta. Učiteľ otvorí okno s vnútorným stavom inštancie a umiestni ho tak, aby bolo viditeľné počas behu aplikácie. Potom spustí aplikáciu a pozoruje, ako sa menia hodnoty atribútov `x`, `y` a `rotation` v inštancii triedy **Enemy** pri volaní rôznych metód. Ako sa tieto hodnoty menia pri pohybe (hore, dole, vľavo a vpravo) a otáčaní?

5. Úloha 3.2 - Pridanie detekcie hraníc sveta

Cieľ:

Žiak vie navrhnúť a implementovať postup pohybu pri detekcii okrajov sveta.

Koncepty na diskusiu:

Detekcia okrajov sveta.

Aktivita:

Učiteľ diskutuje so žiakmi o tom, ako môžu určiť, či je predmet na okraji alebo nie. Napríklad, pokiaľ poznáme rozmery stola, na základe polohy (`x`, `y`) sa dá určiť, či je predmet na hrane stola alebo nie.

Učiteľ umiestni inštanciu triedy **Enemy** niekde vo svete (ale nie na okraj), zavolá metódu `IsAtEdge()`. Diskutuje so žiakmi o tom, čo sa stalo. Učiteľ presunie inštanciu na okraj a preskúma výsledok metódy `IsAtEdge()`, ktorá by teraz mala vrátiť `true`.

Učiteľ vysvetlí metódu `isAtEdge()`.

Učiteľ zadá žiakom úlohu 3.2 - pridať do tela metódy `act()` kód, ktorý otočí nepriateľa o 180° volaním vlastnosti `setRotation()`, keď dosiahne okraj sveta.

Učiteľ so žiakmi prediskutuje, ako môže objekt, ktorý dosiahol okraj, pokračovať vo svojom pohybe:

- dozadu (bez otáčania),
- dozadu (s otáčaním).

Učiteľ spolu so žiakmi doplní programový kód metódy `act()`.

Teraz spustite metódu `act()` a diskutujte o tom, čo sa stane s nepriateľom, keď dosiahne okraj sveta.

Commit: [4927c3ff7eb39b51ba2738f2ab500fd6c32e3bb4](https://github.com/4927c3ff7eb39b51ba2738f2ab500fd6c32e3bb4)

5.2. Skúmanie vetvenia pri vývoji hier s Greenfoot - Úplné vetvenie kódu

Tabuľka 8: Skúmanie vetvenia pri vývoji hier s Greenfoot - Úplné vetvenie kódu

<i>Názov</i>	Skúmanie vetvenia pri vývoji hier s Greenfoot - Úplné vetvenie kódu
<i>Výchovno-vzdelávacie ciele a ich opis</i>	Téma zahŕňa neúplné a úplné vetvenie. Zavádzajú sa základy vnímania sveta aktormi. Žiaci budú schopní písať kód s použitím podmienok. Po ukončení tejto témy sa žiaci naučia používať jednoduché príkazy if-else a naučia sa robiť rozhodnutia v kóde, ktoré kontrolujú správanie ich hry. Získajú základné znalosti programovacieho jazyka Java, naučia sa potrebnú syntax a analyzovať a pochopiť kód, čo im pomôže pochopiť, prečo sa ich hra správa určitým spôsobom a ako odstrániť problémy. Žiaci budú môcť vytvoriť vlastné herné projekty s využitím toho, čo sa naučili o objektovo orientovanom programovaní.
<i>Cieľová skupina</i>	Žiaci stredných škôl na kurze OOP4Fun. Základné znalosti programovania vrátane premenných a dátových typov. Žiaci by mali byť oboznámení s programom Greenfoot.
<i>Časové trvanie</i>	<ol style="list-style-type: none">1. Úloha 3.3 - Pridanie tried Direction a Orb (30 minút)2. Vysvetlenie detekcie kolízií (30 minút)3. Úloha 3.4 - Pridanie detekcie kolízií (10 minút)4. Úloha 3.5 - Predvídanie pohybu nepriateľa na vlastnom usporiadaní (15 minút)5. Úloha 3.6 - Predvídanie pohybu nepriateľa pri špecifickom usporiadaní (15 minút)6. Vysvetlenie kódu: Úplné vetvenie (15 minút)7. Úloha 3.7 – Použitie úplného vetvenia pri detekcii kolízií (20 minút)8. Úloha 3.8 - Predvídanie pohybu nepriateľa na základe predchádzajúcich nastavení (30 minút)9. Opakovanie teórie (5 minút)
<i>Materiály a zdroje</i>	<ul style="list-style-type: none">• Učebnica z projektu OOP4Fun.• Zdroje z projektu OOP4Fun.• Zdrojový kód projektu z Github/Gitlab.• Internetové zdroje.

<p><i>Popis</i></p>	<p>Hodina sa začína 30-minútovým cvičením na pridanie tried Direction a Orb. To pomáha žiakom pochopiť objektovo orientovanú povahu jazyka Java a dôležitosť správneho štruktúrovania kódu. Nasleduje 20-minútové vysvetlenie kódu zamerané na koncept detekcie kolízií, ktorý pomáha žiakom pochopiť interakcie objektov v rámci ich herného prostredia. Potom žiaci strávia 10 minút pridávaním detekcie kolízií do svojho projektu v ďalšej produkčnej úlohe, pričom využijú svoje znalosti vetvenia.</p> <p>Nasledujú výskumné úlohy v trvaní 15 minút, v ktorých majú žiaci za úlohu predvídať pohyb nepriateľa vo vlastnej aj zložitej zostave, čo posilňuje ich schopnosti riešiť problémy a analytické zručnosti. Ďalšia 15-minútová diskusia je zameraná na úplné vetvenie, čím sa zabezpečuje, aby žiaci vedeli rozlišovať medzi neúplnými a úplnými štruktúrami kódu.</p> <p>20-minútová produkčná úloha zapája žiakov do používania úplného vetvenia s detekciou kolízií, pričom cieľom je upevniť ich učivo aplikovaním komplexných konceptov v praktickom prostredí. Hodinu uzatvára náročná 30-minútová investigatívna úloha, v ktorej žiaci opäť predpovedajú pohyb nepriateľa, tentoraz s pridanými skúsenosťami z predchádzajúcich úloh, čím uplatňujú svoje kumulované vedomosti.</p> <p>Vyučovanie sa končí 5-minútovým teoretickým zopakovaním naučených pojmov.</p>
<p><i>Hodnotenie</i></p>	<p>Táto aktivita umožní učiteľom poskytovať formatívnu spätnú väzbu na základe diskusií a monitorovania obrátenej triedy a tímovej práce žiakov.</p> <p>Vzájomné hodnotenie sa bude vykonávať online ako súčasť domácej úlohy. Žiakom sa tak pripomenú dôležité aspekty cvičenia, donútia sa kriticky posúdiť prácu ostatných žiakov, získajú prehľad o dobrých či menej dobrých riešeniach svojich kolegov a pod. a zvýši sa celkové dosiahnutie výsledkov vzdelávania.</p> <p>Pri práci v tímovom projekte, na ktorom žiaci pracujú, sa budú využívať aj tieto výsledky vzdelávania a poznatky.</p>
<p><i>Šírenie výsledkov</i></p>	<p>Na šírenie výsledkov medzi učiteľmi a spolužiakmi sa použije bežné nastavenie Learning management systému (napr. Moodle). Žiaci môžu pokračovať v diskusii na danú tému na fóre, ktoré im je poskytnuté prostredníctvom nástroja na Learning management.</p>

5.2.2. Príručka učiteľa na prípravu vyučovacej hodiny

1. Úloha 3.3 - Pridanie tried `Direction` a `Orb`

Cieľ:

Žiaci pochopia, ako pridať novú triedu do projektu.

Koncepty na diskusiu:

Trieda.

Aktivita:

Učiteľ zadá žiakom úlohu 3.3. Učiteľ sleduje činnosť žiakov a na záver požiada jedného žiaka, aby prezentoval svoju prácu. Žiak opíše a predstaví svoju prácu.

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

2. Vysvetlenie detekcie kolízií

Cieľ:

Vysvetlenie detekcie kolízií.

Koncepty na diskusiu:

Detekcia kolízií.

Aktivita:

Učiteľ umiestnil inštanciu triedy `Enemy` do sveta a inštanciu triedy `Direction` do toho istého riadku. Učiteľ pridá kód do metódy `act()` tak, aby sa objekt posunul o jeden krok dopredu.

Učiteľ vysvetlí žiakom, ako určiť, či sa dva alebo viac objektov („postáv“) vo svete nachádzajú na rovnakej pozícii (na rovnakej bunke). Učiteľ vysvetlí metódu `isTouching()`.

Učiteľ a žiaci upravujú metódu `act()` triedy `Enemy` tak, aby zabezpečila, že sa nepriateľ otočí o 90° v smere hodinových ručičiek, keď sa nachádza na tej istej bunke, ktorá obsahuje inštanciu triedy `Direction`.

Učiteľ spolu so žiakmi pozoruje, čo sa deje s atribútom `rotation`.

3. Úloha 3.4 - Pridanie detekcie kolízií

Cieľ:

Žiaci rozumejú ako určiť, či sú dva objekty na rovnakej bunke.

Koncepty na diskusiu:

Aktivita:

Učiteľ zadá žiakom úlohu 3.4. Učiteľ sleduje činnosť žiakov a na záver požiada jedného žiaka, aby prezentoval svoju prácu. Žiak opíše a predstaví svoju prácu.

Commit: [968e6f195e3def25e11bc41b664ba1715f7da11d](#)

4. Úloha 3.5 - Predvídanie pohybu nepriateľa na vlastnom usporiadaní

Cieľ:

Žiaci rozumejú pohybu objektov.

Koncepty na diskusiu:

Pohyb objektov.

Aktivita:

Učiteľ zadá žiakom úlohu 3.5. Učiteľ sleduje činnosť žiakov a na záver požiada jedného žiaka, aby prezentoval svoju prácu. Žiak opíše a predstaví svoju prácu.

5. Úloha 3.6 - Predvídanie pohybu nepriateľa pri špecifickom usporiadaní

Cieľ:

Žiaci rozumejú pohybu objektov.

Koncepty na diskusiu:

Pohyb objektov.

Aktivita:

Učiteľ zadá žiakom úlohu 3.6. Učiteľ sleduje činnosť žiakov a na záver požiada jedného žiaka, aby prezentoval svoju prácu. Žiak opíše a predstaví svoju prácu.

6. Vysvetlenie kódu: Úplné vetvenie

Cieľ:

Učiteľ vysvetlí, kedy a ako používať príkaz **if-else** a **switch**.

Koncepty na diskusiu:

Príkaz **if-else**, vnorené príkazy **if**, príkaz **switch**.

Aktivita:

Učiteľ zadá žiakom úlohu, aby na papier opísali, ako chodec prechádza cez ulicu. Učiteľ sleduje, ako žiaci riešia úlohu. V určitom momente učiteľ zdôrazní žiakom, aby venovali pozornosť tomu, či na mieste križovania ulice je alebo nie je semafor. Ak medzitým, teda skôr, ako to učiteľ zdôrazní, jeden zo žiakov položí túto otázku alebo niečo podobné, nahlas ho pochváli a zdôrazňuje, že je dôležité, aby sme pred programovaním vždy najskôr analyzovali problém a identifikovali všetky prípady, ku ktorým došlo. Tu by žiaci mali pochopiť úplné a neúplné vetvenie.

Učiteľ požiada žiakov, aby umiestnili inštancie triedy **Orb** a **Direction** doľava, doprava alebo doprava – k rôznym hraniciam sveta. Žiaci opíšu „zlé“ správanie inštancie triedy **Enemy**. Učiteľ by mal vysvetliť, že v jednej situácii sú splnené podmienky, keď sa inštancia

triedy **Enemy** dotýka triedy **Orb** alebo **Direction** a dotýka sa okraja. Nakreslením situácie na tabuľu alebo papier žiaci rozpoznajú, že je potrebné úplné a vnorené vetvenie a potom žiaci zmenia kód metódy **act()** inštancie triedy **Enemy**.

7. Úloha 3.7 – Použitie úplného vetvenia pri detekcii kolízií

Cieľ:

Žiaci rozumejú úplnému vetveniu, vnoreným príkazom **if** a príkazu **switch**.

Koncepty na diskusiu:

Vnorené príkazy **if**, príkaz **switch**.

Aktivita:

Učiteľ zadá žiakom úlohu 3.7. Učiteľ sleduje činnosť žiakov a na záver požiada jedného žiaka, aby prezentoval svoju prácu. Žiak opíše a predstaví svoju prácu.

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

8. Úloha 3.8 - Predvídanie pohybu nepriateľa na základe predchádzajúcich nastavení

Cieľ:

Žiaci rozumejú vplyvu vetvenia v metóde **act()** na správanie inštancie triedy **Enemy** a vedia tak predvídať pohyb nepriateľa na základe predchádzajúcich nastavení.

Koncepty na diskusiu:

Správanie objektov.

Aktivita:

Učiteľ ľubovoľne vkladá predmety do sveta a žiaci vysvetľujú ich pohyb a správanie (samostatne alebo vo dvojiciach). Učiteľ zadá žiakom úlohu 3.8.

9. Opakovanie teórie

Cieľ:

Opakovanie teórie v súvislosti s predtým preberanými pojmami.

Koncepty na diskusiu:

Princípy vetvenia.

Aktivita:

V tejto časti učiteľ vykoná opakovanie predtým diskutovaných konceptov.

6. Premenné a výrazy

V rámci tematického celku Premenné a výrazy bolo vytvorených päť metodických materiálov.

6.1. Úvod do premenných a dátových typov v prostredí Greenfoot

Tabuľka 9: Úvod do premenných a dátových typov v prostredí Greenfoot

<i>Názov</i>	Úvod do premenných a dátových typov v prostredí Greenfoot
<i>Výchovno-vzdelávacie ciele a ich opis</i>	Na konci tejto hodiny budú žiaci schopní porozumieť dátovým typom a premenným . Prebrané pojmy sa budú preberať v kontexte vývoja hry, čím sa podporí tvorivé myslenie, tímová práca a zanietený prístup ku programovaniu v prostredí Greenfoot.
<i>Cieľová skupina</i>	Žiaci stredných škôl na kurze OOP4Fun. Základné znalosti programovania vrátane premenných a dátových typov. Žiaci by mali byť oboznámení s programom Greenfoot.
<i>Časové trvanie</i>	<ol style="list-style-type: none">1. Úvod (10 minút)2. Identifikácia premennej (5 minút)3. Typy údajov (15 minút)4. Deklarácia premenných (10 minút)5. Inicializácia premenných (5 minút)
<i>Materiály a zdroje</i>	<ul style="list-style-type: none">• Učebnica z projektu OOP4Fun.• Zdroje z projektu OOP4Fun.• Zdrojový kód projektu z Github/Gitlab.• Internetové zdroje.
<i>Popis</i>	<p>V tomto 45-minútovom metodickom materiáli sa žiaci stredných škôl oboznámia so zásadami programovania týkajúcimi sa dátových typov a premenných z pohľadu vývoja hier pomocou nástroja Greenfoot. Vyučovanie sa začína 10-minútovým úvodom učiteľa, v ktorom sa vytvorí kontext súvisiaci s predchádzajúcimi vyučovacími hodinami, čo vytvára základ pre zavedenie a definovanie premenných.</p> <p>Potom nasleduje 5-minútový blok, počas ktorého žiaci a učitelia skúmajú a určujú premenné pre svoju hru. Vzhľadom na to, že každá premenná</p>

	<p>musí byť určitého typu, v nasledujúcich 15 minútach sa prezentujú rôzne typy údajov.</p> <p>Základné aktivity zahŕňajú 10-minútové cvičenie pod vedením učiteľa, počas ktorého žiaci deklarujú premenné pre svoju hru, pričom sa zdôrazňuje dôležitosť názvov a typov premenných. Potom nasleduje 5-minútové cvičenie týkajúce sa inicializácie premenných, počas ktorého sa definujú hodnoty premenných. V tejto súvislosti sa môže meniť správanie objektov v hre (napr. otáčanie objektu, pohyb objektu).</p> <p>Žiaci budú pokračovať v práci na hre, ktorá bola vysvetlená a začatá na predchádzajúcich hodinách. Výsledkom je, že na konci hodiny sa zavedú nové pojmy týkajúce sa premenných a dátových typov.</p>
<i>Hodnotenie</i>	<p>Gamifikácia predstavuje neformálne hodnotenie, ale zvýši záujem, vnútornú motiváciu a vzdelávacie výstupy celej skupiny.</p>
<i>Šírenie výsledkov</i>	<p>Na šírenie výsledkov medzi učiteľmi a spolužiakmi sa použije bežné nastavenie Learning management systému (napr. Moodle). Žiaci môžu pokračovať v diskusii na danú tému na fóre, ktoré im je poskytnuté prostredníctvom nástroja na Learning management.</p>

6.1.1. Príručka učiteľa na prípravu vyučovacej hodiny

1. Úvod

V úvodnej časti je uvedený kontext súvisiaci s predchádzajúcimi hodinami. Učiteľ zavádza pojem *premenná*.

2. Identifikácia premennej

Cieľ:

Žiaci identifikujú premenné prostredníctvom diskusie s dôrazom na úlohu premenných v programovaní.

Koncepty na diskusiu:

Premenné a hodnoty.

Aktivita:

- Učiteľ zavádza pojem *premenná*.
- Žiaci môžu byť požiadaní, aby preskúmali a určili premenné pre svoju hru.
- O premenných môže diskutovať učiteľ a spolužiaci.
- V tejto časti možno typ premennej vynechať (alebo o nej diskutovať všeobecne).

3. Typy údajov

Cieľ:

Pochopenie koncepcie dátových typov, rozpoznanie ich reálnych aplikácií a zameranie sa na typy premenných potrebné pre vývoj hier.

Koncepty na diskusiu:

Premenné, Dátové typy, Príklady dátových typov v reálnom svete, Typy premenných pre hru.

Aktivita:

- Učiteľ zavedie pojem dátový typ.
- Možno diskutovať o príkladoch z reálneho sveta (napr. celé čísla môžu súvisieť s počtom aktuálne prítomných žiakov, desatinné čísla môžu súvisieť s cenou výrobku, textový typ môže súvisieť s textom okamžitých správ atď.).
- Dátové typy sa posudzujú v kontexte prostredia Greenfoot a programovacieho jazyka Java.
- Podrobná diskusia týkajúca sa typov premenných potrebných pre hru.

4. Deklarácia premenných

Cieľ:

Aplikovať koncepty dátových typov a premenných a deklarovať premenné potrebné pre vývoj hier.

Koncepty na diskusiu:

Premenné, dátové typy, typy premenných pre hru.

Aktivita:

- Dátové typy sa preberajú v kontexte prostredia Greenfoot a programovacieho jazyka Java.
- Učiteľ by mal vysvetliť rozdiel medzi deklaráciou a inicializáciou premenných:
 - Pri deklarácii premennej sa deklaruje premenná určitého dátového typu, ale hodnota môže/nemusí byť prítomná.
 - Možno zaviesť analógiu (napr. označený box pre konkrétny typ cookie, ale bez cookie vo vnútri označeného boxu).
- Deklarácia premenných vyžadovaných v hre.
- Možno zvážiť ďalšie príklady. Napríklad, ak sa uvažuje o metóde `act()`, možno deklarovať premennú na zobrazenie textu.

5. Inicializácia premenných

Cieľ:

Aplikovať pojmy dátových typov a premenných a inicializovať premenné potrebné pre hru.

Koncepty na diskusiu:

Premenné, dátové typy, hodnoty premenných pre hru.

Aktivita:

- Na základe predtým predstavených dátových typov sa predstavia ich hodnoty a rozsahy.
- Hodnoty údajov a rozsahy údajov sa posudzujú v kontexte prostredia Greenfoot a programovacieho jazyka Java.
- Učiteľ by mal vysvetliť rozdiel medzi deklaráciou a inicializáciou premenných:
 - Pri inicializácii premennej sa deklaruje premenná určitého dátového typu a zároveň sa môže inicializovať (ide o tzv. počiatočnú hodnotu) a v nasledujúcom kóde sa môže meniť.
 - Možno zaviesť analógiu (napr. cookie konkrétneho typu sa umiestni do predtým definovaného označeného poľa).
- Inicializácia premenných vyžadovaných v hre.
- Možno zvážiť ďalšie príklady. Napríklad, ak sa uvažuje o metóde `act()`, možno inicializovať premennú na zobrazenie textu.

6.2. Úvod do operátorov a výrazov v prostredí Greenfoot

Tabuľka 10: Úvod do operátorov a výrazov v prostredí Greenfoot

Názov	Úvod do operátorov a výrazov v prostredí Greenfoot
Výchovno-vzdelávacie ciele a ich opis	Na konci tejto hodiny budú žiaci schopní pochopiť koncept operátorov . V tejto časti sa predstavia rôzne typy operátorov (t. j. aritmetické operátory, logické operátory, relačné operátory), ako aj ich príslušné výrazy . Okrem toho sa predstaví objektový výraz a referenčná premenná. Prebrané pojmy sa budú preberať v kontexte vývoja hry, čím sa podporí tvorivé myslenie, tímová práca a zanieteny prístup ku programovaniu v prostredí Greenfoot.
Cieľová skupina	Žiaci stredných škôl na kurze OOP4Fun. Základné znalosti programovania vrátane konceptov iterácie a výberu. Žiaci by mali byť oboznámení s programom Greenfoot.
Časové trvanie	<ol style="list-style-type: none">1. Operátory (15 minút)2. Aritmetické operátory a výrazy (10 minút)3. Logické operátory (15 minút)4. Relačné operátory (10 minút)5. Logické výrazy (10 minút)

	<ol style="list-style-type: none"> 6. Objektové výrazy (5 minút) 7. Referenčná premenná a jej hodnota null (15 minút) 8. Úloha 4.1 - Otáčanie v smere (15 minút)
<i>Materiály a zdroje</i>	<ul style="list-style-type: none"> • Učebnica z projektu OOP4Fun. • Zdroje z projektu OOP4Fun. • Zdrojový kód projektu z Github/Gitlab. • Internetové zdroje.
<i>Popis</i>	<p>Počas tejto 95-minútovej vyučovacej hodiny sa žiaci stredných škôl oboznámia s programovacími konceptmi týkajúcimi sa operátorov a výrazov v kontexte vývoja hier pomocou nástroja Greenfoot. Hodina sa začína 15-minútovým úvodom učiteľa, ktorý poskytuje základné informácie z predchádzajúcich hodín, čo vytvára základ pre zavedenie a definovanie operátorov.</p> <p>Potom nasleduje 10-minútový úsek, počas ktorého sa preberajú aritmetické operátory a výrazy. aritmetické operátory sa používajú na aritmetické operácie. V tejto súvislosti sa prakticky vysvetľujú a diskutujú rôzne operátory a výrazy v prostredí Greenfoot.</p> <p>V ďalšej 15-minútovej časti sa predstavia logické operátory. Ide o logické operátory používané na manipuláciu s logickými hodnotami. Nasleduje 10-minútová sekcia týkajúca sa relačných operátorov, ktoré sa používajú na porovnávanie hodnôt. Na základe predchádzajúcich relácií sa v ďalšom 10-minútovom úseku rozoberajú logické výrazy v kontexte prostredia Greenfoot.</p> <p>Nasledujúca 5-minútová časť sa zameriava na vyjadrenie objektu, zatiaľ čo nasledujúca 15-minútová časť sa zaoberá referenčnými premennými.</p> <p>Nakoniec nasleduje 15-minútové cvičenie pod vedením učiteľa, počas ktorého žiaci riešia úlohy súvisiace s otáčaním v smere v ich hre. Súčasťou tejto časti je aj spätná väzba poskytnutá učiteľom a spolužiakmi.</p> <p>Žiaci pokračujú v práci na projekte hry, ktorú začali v predchádzajúcich hodinách. Následne sa do konca hodiny oboznámia s pojmami týkajúcimi sa operátorov a výrazov.</p>
<i>Hodnotenie</i>	<p>Gamifikácia predstavuje neformálne hodnotenie, ale zvýši záujem, vnútornú motiváciu a výsledky vzdelávania celej skupiny.</p> <p>Stav projektu otvára možnosti pre domáce úlohy. V tejto súvislosti možno zaviesť ďalšie triedy, výrazy a hodnoty na dosiahnutie ďalšieho</p>

	správania. Tieto koncepty možno so žiakmi prediskutovať a príslušnú realizáciu zadať ako domácu úlohu.
Šírenie výsledkov	Na šírenie výsledkov medzi učiteľmi a spolužiakmi sa použije bežné nastavenie Learning management systému (napr. Moodle). Žiaci môžu pokračovať v diskusii na danú tému na fóre, ktoré im je poskytnuté prostredníctvom nástroja na Learning management.

6.2.1. Príručka učiteľa na prípravu vyučovacej hodiny

1. Operátory

Cieľ:

Žiaci pochopia koncepcie operátorov, ich prepojenie s reálnymi prípadmi a zoznámenie sa s rôznymi typmi operátorov.

Koncepty na diskusiu:

Premenné, dátové typy, operátory, príklady operátorov z reálneho sveta.

Aktivita:

Učiteľ zavedie pojem *operátor*. Učiteľ by mal tento pojem priblížiť žiakom pomocou príkladov z reálneho života (napr. nákup produktov na trhu). Potom učiteľ predstaví rôzne typy operátorov.

2. Aritmetické operátory a výrazy

Cieľ:

Pochopenie konceptu aritmetických operátorov, ich prepojenie s reálnymi príkladmi a zoznámenie sa s rôznymi aritmetickými operátormi.

Koncepty na diskusiu:

Premenné, dátové typy, operátory, príklady aritmetických operátorov z reálneho sveta.

Aktivita:

Učiteľ môže vysvetliť operátory, ktoré už pozná z iných predmetov (napr. matematické a aritmetické operátory). Tieto operátory sa rozoberajú v kontexte prostredia Greenfoot a programovacieho jazyka Java, Učiteľ rozoberá rôzne pojmy: operátor, operand, precedencia operátora. Ďalšie príklady môžu zahŕňať definovanie lokálnych premenných na získanie a manipuláciu s polohou x a y entity, čím sa zmení jej poloha zvýšením hodnôt premenných.

3. Logické operátory

Cieľ:

Pochopenie konceptu logických operátorov, ich prepojenie s reálnymi prípadmi a zoznámenie sa s rôznymi logickými operátormi.

Koncepty na diskusiu:

Premenné, dátové typy, operátory, príklady boolovských operátorov z reálneho sveta.

Aktivita:

Učiteľ môže vysvetliť operátory už známe z iných predmetov (napr. matematické a logické operátory). Tieto operátory sa preberajú v kontexte prostredia Greenfoot a programovacieho jazyka Java. Učiteľ rozoberie rôzne pojmy: operátor, operand, precedencia operátora. Ďalšie príklady môžu zahŕňať definovanie lokálnych premenných na kontrolu, či sa poloha x entity rovná jej polohe y, použitie logického operátora na určenie, či sa entita nachádza na diagonále.

4. Relačné operátory

Cieľ:

Pochopenie konceptu relačných operátorov, ich prepojenie s reálnymi scenármi a zoznámenie sa s rôznymi relačnými operátormi.

Koncepty na diskusiu:

Premenné, dátové typy, operátory, príklady operátorov z reálneho sveta.

Aktivita:

Učiteľ môže vysvetliť operátory, ktoré už pozná z iných kurzov (napr. matematické a matematické relačné operátory). Tieto operátory sa preberajú v kontexte prostredia Greenfoot a programovacieho jazyka Java. Učiteľ rozoberie rôzne pojmy: operátor, operand, precedencia operátora. Ďalšie príklady môžu zahŕňať definovanie lokálnych premenných na kontrolu, či je pozícia y jednej entity pod pozíciou inej entity, použitie relačných operátorov na určenie pozičných vzťahov medzi entitami.

5. Logické výrazy

Cieľ:

Pochopenie konceptu logických výrazov, ich prepojenie s reálnymi prípadmi a zoznámenie sa s rôznymi logickými výrazmi.

Koncepty na diskusiu:

Premenné, dátové typy, operátory, príklady logických výrazov z reálneho sveta.

Aktivita:

Učiteľ dokáže vysvetliť logické výrazy v kontexte predtým prezentovaných operátorov. Tieto výrazy sa preberajú v kontexte prostredia Greenfoot a programovacieho jazyka Java. Učiteľ rozoberie operátor, operand a prioritu operátorov v kontexte logických výrazov.

Môže sa zamyslieť nad ďalšími príkladmi. Napríklad, logické výrazy možno použiť na overenie, či sa pozícia entity nachádza vo vnútri definovaného rozmeru arény v hre.

6. Objektové výrazy

Cieľ:

Pochopenie konceptu objektových výrazov, ich prepojenie s reálnymi prípadmi a zoznámenie sa s rôznymi objektovými výrazmi.

Koncepty na diskusiu:

Premenné, dátové typy, operátory, príklady objektových výrazov z reálneho sveta.

Aktivita:

Učiteľ dokáže vysvetliť vyjadrenie objektu v kontexte objektovo orientovaného návrhu. Tieto výrazy sa rozoberajú v kontexte prostredia Greenfoot a programovacieho jazyka Java. Učiteľ rozoberie operátor, operand, prioritu operátorov a pretypovanie objektov v kontexte objektových výrazov. Možno preskúmať ďalšie príklady. Napríklad porovnať referencie na dve objektové entity a skontrolovať, či sa prekrývajú.

7. Referenčná premenná a jej hodnota null

Cieľ:

Pochopenie konceptu referenčných premenných, ich prepojenie s reálnymi prípadmi a ich aplikácia pri vývoji hry.

Koncepty na diskusiu:

Premenné, dátové typy, operátory, príklady referenčných premenných z reálneho sveta.

Aktivita:

Učiteľ vie vysvetliť referenčné premenné v kontexte objektovo orientovaného návrhu. Tieto referenčné premenné sa preberajú v kontexte prostredia Greenfoot a programovacieho jazyka Java. učiteľ by mal vysvetliť referenčnú hodnotu `null`.

8. Úloha - Otáčanie v smere

Cieľ:

Pochopenie konceptu premenných, dátových typov, operátorov, výrazov a ich využitie pri vývoji hry.

Koncepty na diskusiu:

Premenné, dátové typy, operátory, výrazy.

Aktivita:

Učiteľ by mal prediskutovať metódu `Enemy.act()`. Učiteľ vysvetlí, ako používať lokálne premenné v kóde, napríklad použiť premennú „`rotation`“. Učiteľ by mal opísať rozdiel medzi `this.rotation` a `rotation`. Učiteľ by mal opísať správanie metódy

`getOneIntersectingObject(_cls_)`. Používa sa a inštancia sa ukladá do správnej lokálnej premennej (vyžaduje sa pretypovanie objektu). Ak neexistuje žiadny objekt pretínajúci sa s objektom, vráti hodnotu `null`. Na základe ohodnotenia logického výrazu sa vykoná príslušná činnosť. O výsledkoch diskutuje učiteľ a spolužiaci.

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

6.3. Úvod do konštruktorov v prostredí Greenfoot

Tabuľka 11: Úvod do konštruktorov v prostredí Greenfoot

<i>Názov</i>	Úvod do konštruktorov v prostredí Greenfoot
<i>Výchovno-vzdelávacie ciele a ich opis</i>	Na konci tejto hodiny budú žiaci schopní porozumieť konceptu konštruktorov . Na hodine sa predstavia základné teoretické pojmy týkajúce sa konštruktorov, ako aj rôzne vysvetlenia kódu a úlohy. Prebrané pojmy sa budú preberať v kontexte vývoja hry, čím sa podporí tvorivé myslenie, tímová práca a zaniatený prístup ku programovaniu v prostredí Greenfoot.
<i>Cieľová skupina</i>	Žiaci stredných škôl na kurze OOP4Fun. Základné znalosti programovania vrátane konceptov iterácie a výberu. Žiaci by mali byť oboznámení s programom Greenfoot.
<i>Časové trvanie</i>	<ol style="list-style-type: none"> 1. Základné princípy konštruktorov (10 minút) 2. Vysvetlenie kódu (20 minút) 3. Úloha 4.2 - Premenovanie triedy MyWorld na Arena (5 minút) 4. Úloha 4.3 - Vytvorenie usporiadania arény (30 minút)
<i>Materiály a zdroje</i>	<ul style="list-style-type: none"> • Učebnica z projektu OOP4Fun. • Zdroje z projektu OOP4Fun. • Zdrojový kód projektu z Github/Gitlab. • Internetové zdroje.
<i>Popis</i>	<p>Počas tejto 65-minútovej hodiny sa žiaci stredných škôl oboznámia s pojmami týkajúcimi sa konštruktorov v kontexte vývoja hier pomocou nástroja Greenfoot. Vyučovanie začína 10-minútovým úvodom učiteľa, ktorý poskytuje základné teoretické pojmy o konštruktoroch.</p> <p>Potom nasleduje 20-minútový úsek, počas ktorého sa prakticky vysvetľujú a diskutujú rôzne príklady kódu v prostredí Greenfoot.</p>

	<p>Ďalšia 5-minútová časť relácie sa týka úlohy: Predtým definovaná trieda MyWorld by sa mala premenovať. V tejto súvislosti by sa mal definovať nový názov, t. j. názov triedy by mal byť Arena. Dôležité je poznamenať, že konštruktor triedy by sa mal tiež premenovať.</p> <p>Nakoniec nasleduje 30-minútové cvičenie pod vedením učiteľa, počas ktorého žiaci riešia úlohu týkajúcu sa rozloženia triedy A Arena. Súčasťou tohto cvičenia je spätná väzba poskytnutá učiteľom a spolužiakmi.</p> <p>Žiaci budú pokračovať v práci na hernom projekte začatom v predchádzajúcich sedeniach. Následne sa do konca hodiny oboznámia s pojmami súvisiacimi s konštruktormi.</p>
<i>Hodnotenie</i>	<p>Gamifikácia predstavuje neformálne hodnotenie, ale zvýši záujem, vnútornú motiváciu a výsledky vzdelávania celej skupiny.</p> <p>Stav projektu otvára možnosti pre domáce úlohy. V tejto súvislosti možno zaviesť ďalšie triedy, výrazy a hodnoty na dosiahnutie ďalšieho správania. Tieto koncepty možno so žiakmi prediskutovať a príslušnú realizáciu zadať ako domácu úlohu.</p>
<i>Šírenie výsledkov</i>	<p>Na šírenie výsledkov medzi učiteľmi a spolužiakmi sa použije bežné nastavenie Learning management systému (napr. Moodle). Žiaci môžu pokračovať v diskusii na danú tému na fóre, ktoré im je poskytnuté prostredníctvom nástroja na Learning management.</p>

6.3.1. Príručka učiteľa na prípravu vyučovacej hodiny

1. Základné princípy konštruktorov

Cieľ:

Žiaci porozumejú konceptu konštruktorov.

Koncepty na diskusiu:

Konštruktory, triedy, objekty, kľúčové slová: **super, new, this**.

Aktivita:

Učiteľ predstaví pojem konštruktor v kontexte pojmov trieda a objekt v objektovo orientovanom programovaní. Konštruktory sa používajú na inicializáciu konkrétnej inštancie triedy.

2. Vysvetlenie kódu

Cieľ:

Pochopenie konceptu konštruktorov, ich prepojenie s reálnymi situáciami.

Koncepty na diskusiu:

Konštruktory, triedy, objekty, metódy, parametre, kľúčové slová: **super**, **new**, **this**, príklady konštruktorov z reálneho sveta.

Aktivita:

Učiteľ by mal diskutovať o konštruktoroch v kontexte konceptov triedy a objektu OOP: konštruktory sa používajú na inicializáciu inštancií triedy. Okrem toho sú konštruktory vždy vyvolané a môžu byť definované buď implicitne, alebo explicitne. Existujú predvolené konštruktory (ktoré sú implicitne definované), ako aj parametrické a bezparametrické konštruktory (ktoré sú explicitne definované programátorom). Rozdiely medzi parametrickými a bezparametrickými konštruktormi by mali byť tiež diskutované. Aby sa tento pojem lepšie priblížil žiakom, učiteľ by mal používať príklady z reálneho života..

3. [Úloha 4.2 - Premenovanie triedy MyWorld na Arena](#)

Cieľ:

Premenovanie triedy **MyWorld**.

Koncepty na diskusiu:

Konštruktory, triedy, objekty.

Aktivita:

Predtým definovaná trieda **MyWorld** by sa mala premenovať. V tejto súvislosti by sa mal zvoliť nový názov, konkrétne **Arena**. Okrem toho by sa mal premenovať aj konštruktor triedy z **MyWorld()** na **Arena()**.

Commit: [aaf73c9bfd9f76a2a1e504f5e78d2976f1cada12](#)

4. [Úloha 4.3 - Vytvorenie usporiadania arény](#)

Cieľ:

Pochopenie konceptu konštruktorov a ich prepojenie so situáciou v hre.

Koncepty na diskusiu:

Konštruktory, triedy, objekty, metódy, parametre, kľúčové slová: **super**, **new**, **this**.

Aktivita:

V tejto aktivite by sa malo vytvoriť vlastné rozloženie pre Arénu. Vlastné rozloženie by sa malo zabezpečiť v rámci konštruktoru triedy **Arena**: mala by sa pridať jedna inštancia triedy **Enemy**, jedna inštancia triedy **Orb** a aspoň jedna inštancia triedy **Direction**. Po deklarovaní a inicializácii premenných pre jednotlivé objekty by sa týmto objektom mali

priradiť vlastnosti vyvolaním ich príslušných metód. Nakoniec by sa tieto objekty mali začleniť do arény vyvolaním metódy `addObject(Actor)`.

Commit: [8b105ea2eaf697f08c321efe687ddd31e2d0a041](https://github.com/8b105ea2eaf697f08c321efe687ddd31e2d0a041)

6.4. Úvod do atribútov v prostredí Greenfoot

Tabuľka 12: Úvod do atribútov v prostredí Greenfoot

<i>Názov</i>	Úvod do atribútov v prostredí Greenfoot
<i>Výchovno-vzdelávacie ciele a ich opis</i>	Na konci tejto hodiny budú žiaci schopní porozumieť konceptu atribútov . Na hodine sa predstavia základné teoretické pojmy týkajúce sa atribútov, ako aj rôzne vysvetlenia kódu a úlohy. Skúmané pojmy sa budú preberať v kontexte vývoja hry, čím sa podporí tvorivé myslenie, tímová práca a zanieteny prístup ku programovaniu v prostredí Greenfoot.
<i>Cieľová skupina</i>	Žiaci stredných škôl na kurze OOP4Fun. Základné znalosti programovania vrátane konceptov iterácie a výberu. Žiaci by mali byť oboznámení s programom Greenfoot.
<i>Časové trvanie</i>	<ol style="list-style-type: none"> 1. Úloha 4.4 - Identifikácia problému s pohybom a návrh riešenia (30 minút) 2. Atribúty (10 minút) 3. Parametre konštruktorov (10 minút) 4. Úloha 4.5 - Atribút <code>moveDelay</code> triedy <code>Enemy</code> (20 minút) 5. Úloha 4.6 - Pohyb nepriateľov s dodržaním oneskorenia (30 minút)
<i>Materiály a zdroje</i>	<ul style="list-style-type: none"> • Učebnica z projektu OOP4Fun. • Zdroje z projektu OOP4Fun. • Zdrojový kód projektu z Github/Gitlab. • Internetové zdroje.
<i>Popis</i>	<p>Počas tejto 100-minútovej vyučovacej hodiny sa žiaci stredných škôl oboznámia s pojmami týkajúcimi sa atribútov v kontexte vývoja hier pomocou nástroja Greenfoot. Hodina sa začína 30-minútovou úlohou týkajúcou sa problémov súvisiacich s pohybom a možných riešení.</p> <p>Na základe predchádzajúcej úlohy sa v ďalšej 10-minútovej časti predstavia pojmy súvisiace s atribútmi. Nasleduje 10-minútový úsek,</p>

	<p>počas ktorého sa vysvetľujú a diskutujú parametre konštruktérov.</p> <p>Ďalšie 20-minútové cvičenie pod vedením učiteľa súvisí s úlohou. Definuje sa nový atribút týkajúci sa pohybu v triede Enemy. Okrem toho sa definuje parametrický konštruktor. Súčasťou tejto časti je spätná väzba poskytnutá učiteľom a rovesníkmi.</p> <p>Nakoniec sa v poslednej 30-minútovej časti riadenej učiteľom implementuje pohyb nepriateľov. V tejto súvislosti sa aktualizuje metóda <code>act()</code>. V tejto fáze je zahrnutá spätná väzba poskytnutá učiteľom a spolužiakmi.</p> <p>Žiaci budú pokračovať v práci na hernom projekte začatom v predchádzajúcich hodinách. Následne sa do konca hodiny oboznámia s pojmami súvisiacimi s atribútmi.</p>
<i>Hodnotenie</i>	<p>Gamifikácia predstavuje neformálne hodnotenie, ale zvýši záujem, vnútornú motiváciu a výsledky vzdelávania celej skupiny.</p> <p>Stav projektu otvára možnosti pre domáce úlohy. V tejto súvislosti možno zaviesť ďalšie triedy, výrazy a hodnoty na dosiahnutie ďalšieho správania. Tieto koncepty možno so žiakmi prediskutovať a príslušnú realizáciu zadať ako domácu úlohu.</p>
<i>Šírenie výsledkov</i>	<p>Na šírenie výsledkov medzi učiteľmi a spolužiakmi sa použije bežné nastavenie Learning management systému (napr. Moodle). Žiaci môžu pokračovať v diskusii na danú tému na fóre, ktoré im je poskytnuté prostredníctvom nástroja na Learning management.</p>

6.4.1. Príručka učiteľa na prípravu vyučovacej hodiny

1. Úloha 4.4 - Identifikácia problému s pohybom a návrh riešenia

Cieľ:

Žiaci dokážu nájsť problém súvisiaci s pohybom a možných riešení v kontexte konštruktérov a atribútov.

Koncepty na diskusiu:

Konštruktory, atribúty, metódy.

Aktivita:

Učiteľ by mal vysvetliť, že nepriateľ (inštancia triedy **Enemy**) sa v súčasnosti pohybuje dvoma bunkami naraz, čo spôsobuje problémy s jeho pohybom. Na vyriešenie tohto problému možno rýchlosť nepriateľa modelovať inak. Inštancia triedy **Enemy** sa teraz bude

vždy pohybovať po jednej bunke. Okrem toho možno definovať nový atribút s názvom `moveDelay`, ktorý spôsobí, že sa inštancia triedy `Enemy` začne pohybovať až po uplynutí určitého počtu volaní metódy `act()`.

2. Atribúty

Cieľ:

Pochopenie konceptu atribútov.

Koncepty na diskusiu:

Konštruktory, triedy, objekty, atribúty.

Aktivita:

Učiteľ predstaví koncept atribútov v kontexte konceptov triedy a objektu v objektovo orientovanom programovaní.

3. Parametre konštruktorov

Cieľ:

Pochopenie konceptu parametrov konštruktora.

Koncepty na diskusiu:

Konštruktory, triedy, objekty, atribúty, parametre, kľúčové slová: **super**, **new**, **this**.

Aktivita:

Učiteľ uvedie pojem parametre konštruktora v kontexte pojmov trieda a objekt v objektovo orientovanom programovaní.

4. Úloha 4.5 - Atribút `moveDelay` triedy `Enemy`

Cieľ:

Pochopenie pojmu atribútov a parametrov konštruktora.

Koncepty na diskusiu:

Konštruktory, triedy, objekty, atribúty, parametre, kľúčové slová: **super**, **new**, **this**.

Aktivita:

Do triedy `Enemy` sa pridá nový atribút s názvom `moveDelay` typu `int`. Bude tiež definovaný parametrický konštruktor na inicializáciu tohto atribútu, pričom atribút bude nastavený na hodnotu uvedenú v parametri. Kód v triede `Arena` sa zodpovedajúcim spôsobom upraví.

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

5. Úloha 4.6 - Pohyb nepriateľov s dodržaním oneskorenia

Cieľ:

Pochopenie pojmu atribútov a parametrov konštruktora.

Koncepty na diskusiu:

Konštruktory, triedy, objekty, atribúty, parametre, kľúčové slová: **super**, **new**, **this**.

Aktivita:

Metóda `act()` triedy `Enemy` bude aktualizovaná tak, aby sa inštancia triedy `Enemy` pohla až po `moveDelay` počte volaní metódy. Okrem toho sa zavedie nový atribút s názvom `nextMoveCounter` typu `int`, ktorý sa v konštruktore inicializuje na hodnotu 0. Metóda `act()` sa upraví tak, aby volala `this.move(1)` len vtedy, keď `nextMoveCounter` dosiahne hodnotu 0. Po vykonaní pohybu sa `nextMoveCounter` nastaví na hodnotu `moveDelay`. Ak sa inštancia triedy `Enemy` nemôže pohnúť, pretože `nextMoveCounter` ešte nedosiahol hodnotu 0, tak sa `nextMoveCounter` sa zníži o 1.

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

6.5. Úvod do preťažovania konštruktov v prostredí Greenfoot

Tabuľka 13: Úvod do preťažovania konštruktov v prostredí Greenfoot

Názov	Úvod do preťažovania konštruktov v prostredí Greenfoot
Výchovno-vzdelávacie ciele a ich opis	Na konci tejto hodiny budú žiaci schopní porozumieť konceptu preťažovania konštruktov. Na hodine sa predstavia základné teoretické pojmy týkajúce sa preťažovania konštruktov, ako aj rôzne vysvetlenia kódu a úlohy. Skúmané pojmy sa budú preberať v kontexte vývoja hry, čím sa podporí tvorivé myslenie, tímová práca a nadšený prístup ku programovaniu v prostredí Greenfoot.
Cieľová skupina	Žiaci stredných škôl na kurze OOP4Fun. Základné znalosti programovania vrátane konceptov iterácie a výberu. Žiaci by mali byť oboznámení s programom Greenfoot.
Časové trvanie	<ol style="list-style-type: none">1. Základné princípy preťažovania konštruktov (5 minút)2. Úloha 4.7 - Parametrický konštruktor triedy <code>Direction</code> (25 minút)3. Úloha 4.8 - Preťaženie konštruktov v triede <code>Direction</code> (25 minút)4. Opakovanie teórie (20 minút)

<p><i>Materiály a zdroje</i></p>	<ul style="list-style-type: none"> • Učebnica z projektu OOP4Fun. • Zdroje z projektu OOP4Fun. • Zdrojový kód projektu z Github/Gitlab. • Internetové zdroje.
<p><i>Popis</i></p>	<p>Počas tejto 75-minútovej vyučovacej hodiny sa žiaci stredných škôl oboznámia s konceptmi súvisiacimi s preťažovaním konštruktorov v kontexte vývoja hier pomocou nástroja Greenfoot. Vyučovanie sa začína 5-minútovým úvodom do základných pojmov preťažovania konštruktorov.</p> <p>Ďalšia 25-minútová časť pod vedením učiteľa sa týka úlohy, v ktorej sa definuje parametrický konštruktor v triede Direction. V ďalšej 25-minútovej úlohe sa definuje preťažený konštruktor v triede Direction. Súčasťou tejto časti je spätná väzba poskytnutá učiteľom a spolužiakmi.</p> <p>Nakoniec sa v poslednej 20-minútovej časti pod vedením učiteľa vykonáva teoretická revízia súvisiaca s predtým preberanými pojmami (t. j. premenné, výrazy, operátory, konštruktory, atribúty a preťažovanie konštruktorov).</p>
<p><i>Hodnotenie</i></p>	<p>Gamifikácia predstavuje neformálne hodnotenie, ale zvýši záujem, vnútornú motiváciu a výsledky vzdelávania celej skupiny.</p> <p>Stav projektu otvára možnosti pre domáce úlohy. V tejto súvislosti možno zaviesť ďalšie triedy, výrazy a hodnoty na dosiahnutie ďalšieho správania. Tieto koncepty možno so žiakmi prediskutovať a príslušnú realizáciu zadať ako domácu úlohu.</p>
<p><i>Šírenie výsledkov</i></p>	<p>Na šírenie výsledkov medzi učiteľmi a spolužiakmi sa použije bežné nastavenie Learning management systému (napr. Moodle). Žiaci môžu pokračovať v diskusii na danú tému na fóre, ktoré im je poskytnuté prostredníctvom nástroja na Learning management.</p>

6.5.1. Príručka učiteľa na prípravu vyučovacej hodiny

1. Základné princípy preťažovania konštruktorov

Cieľ:

Žiaci analyzovať preťažovanie konštruktorov.

Koncepty na diskusiu:

Konštruktory.

Aktivita:

Preberajú sa koncepty preťažovania konštruktorov.

2. Úloha 4.7 - Parametrický konštruktor triedy `Direction`

Cieľ:

Definícia parametrického konšuktora triedy `Direction` v kontexte vývoja hry.

Koncepty na diskusiu:

Konštruktory, parametre, atribúty, parametrické konštruktory.

Aktivita:

V tejto časti je pre triedu `Direction` definovaný parametrický konštruktor s jediným parametrom `rotation` typu `int`. V tele konšuktora by sa vytvorená inštancia mala otočiť na základe hodnoty tohto parametra. Kód v triede `Arena` by sa mal zodpovedajúcim spôsobom aktualizovať.

Commit: [3c4b9ef57ab17bac2a0abc7fc5e76ea4b6e27e4b](#).

3. Úloha 4.8 - Preťaženie konštruktorov v triede `Direction`

Cieľ:

Definícia preťaženia konšuktora triedy `Direction` v kontexte vývoja hry.

Koncepty na diskusiu:

Konštruktory, preťaženie konštruktorov.

Aktivita:

V tejto časti je v triede `Direction` definovaný preťažený konštruktor. Pridá sa bezparametrický konštruktor a v jeho tele sa vyvolá parametrický konštruktor s argumentom `rotation` nastaveným na 0. Kód v triede `Arena` by sa mal zodpovedajúcim spôsobom aktualizovať a podľa možnosti použiť bezparametrickú verziu konšuktora triedy `Direction`.

Commit: [1e67e67523c66acea4e93363c9a3173302f424c8](#).

4. Opakovanie teórie

Cieľ:

Opakovanie teórie v súvislosti s predtým preberanými pojmami.

Koncepty na diskusiu:

Premenné, výrazy, operátory, konštruktory, atribúty, preťažovanie konštruktorov.

Aktivita:

V tejto časti sa vykoná opakovanie predtým diskutovaných konceptov.

7. Asociácia

V rámci tematického celku Asociácia boli vytvorené štyri metodické materiály.

7.1. Greenfoot objekty a ich spolupráca: preskúmanie metód a asociácií

Tabuľka 14: Greenfoot objekty a ich spolupráca: preskúmanie metód a asociácií

<i>Názov</i>	Greenfoot objekty a ich spolupráca: preskúmanie metód a asociácií
<i>Výchovno-vzdelávacie ciele a ich opis</i>	<p>Na konci lekcie by mali žiaci dobre pochopiť, ako môžu objekty navzájom interagovať. Inštancia triedy Enemy interaguje s inými objektmi, najmä s triedou Orb, v prostredí Greenfoot. Mali by preukázať zručnosť pri vytváraní a volaní metód v rámci tried jazyka Java, konkrétne implementovať a testovať metódy Arena.respawn(Enemy) a Orb.hit(Enemy). Okrem toho by žiaci mali pochopiť a efektívne spravovať atribúty tried vrátane definovania a používania atribútov Enemy.attack a Orb.hp. Žiaci by mali ovládať zapuzdrovanie údajov v rámci triedy, čo demonštrujú vytvorením getterov (t.j. metód, ktoré vrátia príslušným spôsobom hodnotu atribútu, resp. atribútov), ako napríklad <code>getEnemy.attack()</code>, a pochopiť dôležitosť zapuzdrovania údajov pre bezpečný a udržiavateľný kód. Mali by pochopiť a implementovať odovzdávanie správ medzi objektmi a zabezpečiť, aby inštancie tried efektívne komunikovali pri vykonávaní herných akcií. Okrem toho by žiaci mali používať techniky volania metód na riešenie úloh vývoja interaktívnych hier, pričom by mali efektívne využívať syntax a parametre potrebné na volanie metód.</p>
<i>Cieľová skupina</i>	Žiaci stredných škôl na kurze OOP4Fun. Základné znalosti programovania vrátane konceptov iterácie a výberu. Žiaci by mali byť oboznámení s programom Greenfoot.
<i>Časové trvanie</i>	<ol style="list-style-type: none">1. Úloha 5.1 - Diskusia o tom, čo by sa malo stať, keď nepriateľ dosiahne cieľové miesto (10 minút)2. Úloha 5.2 - Diskusia o tom, ako by mala inštancia triedy Enemy interagovať s príslušnými objektmi pomocou správ pri zásahu inštancie triedy Orb (15 minút)3. Úloha 5.3 - Atribúty attack triedy Enemy a hp triedy Orb (10 minút)4. Metóda (15 minút)5. Úloha 5.4 - Získanie hodnoty atribútu attack triedy Enemy (5 minút)

	<p>6. Úloha 5.5 - Vytvorenie a testovanie metódy <code>respawn(Enemy)</code> triedy <code>Arena</code> (10 minút)</p> <p>7. Úloha 5.6 - Vytvorenie a testovanie metódy <code>hit(Enemy)</code> triedy <code>Orb</code> (10 minút)</p>
<p><i>Materiály a zdroje</i></p>	<ul style="list-style-type: none"> • Učebnica z projektu OOP4Fun. • Zdroje z projektu OOP4Fun. • Zdrojový kód projektu z Github/Gitlab. • Internetové zdroje.
<p><i>Popis</i></p>	<p>V tejto 75-minútovej lekcii sa žiaci stredných škôl zoznámia s prístupmi ako interakcia objektov, vytvárania metód a práca s atribútmi v prostredí Greenfoot. Cieľom tejto lekcie je zlepšiť pochopenie žiakov o tom, ako objekty komunikujú a vzájomne sa ovplyvňujú, čo je kľúčový aspekt objektovo orientovaného programovania.</p> <p>Hodina sa začína 10-minútovou diskusiou o dynamike medzi objektmi nepriateľov a cieľového miesta, pričom sa diskusia zameria na to, čo by sa malo stať, keď sa nepriateľ dostane k cieľovému miestu. Tým sa žiaci pripravujú pre pochopenie interakcií objektov v kontexte hry.</p> <p>Následne sa žiaci zapoja do 15-minútovej úlohy, v ktorej budú diskutovať o tom, ako by mala inštancia triedy Enemy interagovať s príslušnými objektmi pomocou správ, keď narazí na inštanciu triedy Orb. Táto diskusia zdôrazní dôležitosť komunikácie objektov a odovzdávania správ.</p> <p>Ďalej sa 10-minútová úloha zameria na atribúty Enemy.attack a Orb.hp. Žiaci budú definovať a pochopia tieto atribúty, ktoré sú kľúčové pre riadenie herných mechaník.</p> <p>V nasledujúcich 15 minútach sa žiaci zamerajú na metódy a naučia sa ich vytvárať a implementovať v rámci svojich tried. Táto časť umožní bližšie pochopenie vytvárania a volania metód.</p> <p>V 5-minútovej úlohe budú mať žiaci za úlohu vytvoriť getter pre atribút Enemy.attack, čím zlepšia svoje vedomosti o zapuzdrowaní a získavaní údajov.</p> <p>Ďalších 10 minút bude venovaných vytvoreniu a otestovaniu metódy Arena.respawn(Enemy). Žiaci implementujú túto metódu na spracovanie logiky opätovného zjavenia nepriateľských objektov, čím sa docielia pochopenie funkčnosti a testovania metód.</p> <p>Po tejto úlohe bude nasledovať ďalšia 10-minútová úloha, ktorá bude zahŕňať vytvorenie a testovanie metódy Orb.hit(Enemy), ktorá bude</p>

	<p>spracovávať logiku interakcie, keď nepriateľ zasiahne orb.</p> <p>Počas celej hodiny budú žiaci pracovať samostatne alebo v malých skupinách, čím sa podporí spolupráca a vzájomné učenie. Aktívnou účasťou na diskusiách, kódovacích úlohách a testovaní metód si žiaci rozvinú schopnosti kritického myslenia a riešenia výpočtových problémov.</p> <p>Na záver stretnutia žiaci získajú komplexné znalosti o interakciách objektov, tvorbe metód a správe atribútov v prostredí Greenfoot. Tieto základné zručnosti sú potrebné pre ďalšie projekty vývoja hier a zvýšia ich celkovú zdatnosť v programovaní.</p>
<i>Hodnotenie</i>	Gamifikácia predstavuje neformálne hodnotenie, ale zvýši záujem, vnútornú motiváciu a vzdelávacie výstupy celej skupiny.
<i>Šírenie výsledkov</i>	Na šírenie výsledkov medzi učiteľmi a spolužiakmi sa použije bežné nastavenie Learning management systému (napr. Moodle). Žiaci môžu pokračovať v diskusii na danú tému na fóre, ktoré im je poskytnuté prostredníctvom nástroja na Learning management.

7.1.1. Príručka učiteľa na prípravu vyučovacej hodiny

1. Úloha 5.1 - Diskusia o tom, čo by sa malo stať, keď nepriateľ dosiahne cieľové miesto

Cieľ:

Žiaci vedú diskusiu o očakávanom správaní, keď sa nepriateľ dostane ku cieľovému miestu. Žiaci dokážu diskutovať o dynamike hry a jej výsledkoch, napríklad o poškodení cieľového miesta alebo o ukončení hry.

Koncepty na diskusiu:

Poškodenie cieľového miesta, odstránenie nepriateľa, aktivovanie herných udalostí (napr. zníženie zdravia, prehrávanie zvukových efektov, ukončenie hry).

Aktivita:

Hodina sa začína zopakovaním predtým preberaných pojmov, aby si žiaci osvojili nové vedomosti. Učiteľ zapojí žiakov do diskusie s cieľom objasniť pojmy interakcie objektov, správ a metód v kontexte prostredia Greenfoot.

Žiaci spolupracujú v skupinách a navrhujú možnosti, ktoré môžu nastať, keď nepriateľ dosiahne cieľové miesto v ich hre. Rozvíjajú algoritmy, ako napríklad znižovanie bodov zdravia (HP) cieľového miesta pri kontakte s nepriateľom, a rozoberajú scenáre, v ktorých HP gule môže klesnúť na nulu, čo vedie k ukončeniu hry. Prípadne, ak HP cieľového miesta

zostane nad nulou, dohodnú sa na opätovnom vynorení nepriateľa na inom mieste arény. Zvažujú tiež integráciu ďalších herných udalostí vyvolaných touto interakciou, ako sú zvukové efekty alebo správy na obrazovke. Počas celého sedenia učiteľ vedie diskusiu a nabáda žiakov, aby zosúlادili svoje navrhované algoritmy s konkrétnym scenárom: zabezpečiť, aby sa po dosiahnutí cieľového miesta nepriateľom znížilo HP cieľového miesta, ako už bolo uvedené predtým. Toto vedenie učiteľom pomáha žiakom prakticky aplikovať ich nápady, posilňuje ich chápanie hernej dynamiky a interakcií v rámci ich hry.

2. Úloha 5.2 - Diskusia o tom, ako by mala inštancia triedy **Enemy** interagovať s príslušnými objektmi pomocou správ pri zásahu inštancie triedy **Orb**

Cieľ:

Učiteľ sa formou diskusie so žiakmi zameria na proces pochopenia toho, ako by inštancia triedy **Enemy** mala interagovať s inými objektmi pomocou správ, konkrétne keď zasiahne inštanciu triedy **Orb**.

Koncepty na diskusiu:

Volanie metódy, odovzdávanie parametrov, referencie na objekt.

Aktivita:

Žiaci spolupracujú vo dvojiciach, aby preskúmali, ako by inštancia triedy **Enemy** mala interagovať s inštanciou triedy **Orb** pomocou správ v rámci ich herného scenára. Analyzujú a mapujú postupnosť správ a akcií, ktoré by sa mali rozvinúť, keď nepriateľ zasiahne cieľové miesto. Cieľom tohto cvičenia je prehĺbiť ich pochopenie volania metód, odovzdávania parametrov a referencií na objekty v kontexte vývoja hry. Keď žiaci diskutujú o svojich nápadoch a zdokonaľujú ich, učiteľ ich povedie a usmerní tak, aby postupnosť interakcií medzi objektmi zodpovedala algoritmu, ktorý je rozložený medzi spolupracujúcimi objektmi, ako je uvedené v cieľoch hodiny. Pre lepšie pochopenie v tomto procese môže učiteľ zaviesť a použiť sekvenčný diagram UML na vizuálny opis interakcií medzi triedami **Enemy**, **Orb**, **Arena** a **Greenfoot**. Tento vizuálny nástroj pomáha žiakom lepšie porozumieť priebehu posielania správ a volaní metód, čím posilňuje ich chápanie konceptov objektovo orientovaného programovania a ich aplikácie v programovaní v jazyku Java a prostredí Greenfoot.

3. Úloha 5.3 - Atribúty attack triedy **Enemy** a hp triedy **Orb**

Cieľ:

Učiteľ predstaví atribúty **Enemy.attack** a **Orb.hp** a vysvetlí ich význam pri určovaní výsledku interakcie.

Koncepty na diskusiu:

Atribúty triedy, zapuzdrenie.

Aktivita:

Učiteľ predstaví koncept atribútov tried a zapuzdrenia a vysvetlí, ako môžu atribúty ako **Enemy.attack** a **Orb.hp** predstavovať základné vlastnosti objektov v hre. Žiaci sa naučia definovať a používať tieto atribúty vo svojom kóde na modelovanie sily útoku nepriateľa a bodov zdravia cieľového miesta. Začínajú pridaním nového celočíselného atribútu s názvom **attack** do triedy **Enemy**, pričom do konštruktora zahrnú parameter na inicializáciu tohto atribútu. Podobne pridajú do triedy **Orb** celočíselný atribút s názvom **hp** spolu s parametrickým konštruktorom na nastavenie tejto hodnoty pri vytvorení objektu. Učiteľ vedie žiakov pri úprave kódu v triede **Arena** tak, aby obsahoval tieto nové atribúty. Táto praktická skúsenosť pomáha žiakom pochopiť úlohu atribútov tried pri interakcii objektov a dôležitosť zapuzdrenia pri zachovaní integrity a bezpečnosti kódu.

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

4. Metóda

Cieľ:

Učiteľ poskytne prehľad metód a vysvetlí, ako sa používajú na zapuzdrenie činností a správania v rámci tried.

Koncepty na diskusiu:

Definícia metódy, volanie metódy, parametre, návratové hodnoty.

Aktivita:

Učiteľ začne vysvetľovať pojem metódy ako zapuzdrené činnosti alebo správanie v rámci triedy. Na praktických príkladoch demonštruje syntax a štruktúru definícií metód a ukáže, ako sa metódy vyvolávajú na objektoch. Žiaci sa oboznámia s rôznymi typmi metód vrátane tých, ktoré vykonávajú akcie (metódy typu **void**), a tých, ktoré vrátia hodnoty (metódy typu **return**). Učiteľ vysvetlí, ako sa metódam odovzdávajú parametre, pričom zdôrazní dôležitosť typov a poradia parametrov. Prostredníctvom riadených kódovacích cvičení si žiaci precvičia definovanie metód s rôznymi typmi parametrov a návratových hodnôt a vyvolávanie týchto metód na inšanciách objektov. Skúmajú situácie, v ktorých metódy vykonávajú akcie, modifikujú stavy objektov alebo vrátia špecifické hodnoty, čím si upevňujú svoje chápanie funkčnosti metód v rámci triedy.

5. Úloha 5.4 - Získanie hodnoty atribútu **attack** triedy **Enemy**

Cieľ:

Učiteľ vysvetlí koncept metód **getter** a ich účel pri prístupe k hodnotám atribútov.

Koncepty na diskusiu:

Prístupové metódy - **getter**y, zapuzdrenie.

Aktivita:

Učiteľ na začiatku vysvetlí účel metód typu **getter** a zdôrazní, ako poskytujú kontrolovaný prístup k hodnotám atribútov pri zachovaní zapuzdrenia. Žiaci sa naučia,

aké dôležité je používať gettery na získavanie súkromných hodnôt atribútov, čím sa posilní koncept ochrany údajov v rámci triedy. Učiteľ potom vedie žiakov procesom vytvárania getter metódy pre atribút **attack** v triede **Enemy**. Pomocou praktického prístupu žiaci implementujú metódu getter a zabezpečia, aby vrátila hodnotu atribútu **attack**. Učiteľ demonštruje správnu syntax a štruktúru definovania metódy getter a spôsob jej použitia v rámci kódu na prístup k hodnote atribútu. Na konci aktivity by žiaci mali byť schopní vytvoriť a použiť metódy getter na kontrolovaný prístup k hodnotám atribútov, čím sa zlepší ich chápanie zapuzdrenia a ochrany údajov v objektovo orientovanom programovaní.

Commit: [72b7456ea4cc11416c57d72c89b6a7f7e9266e3e](#)

6. Úloha 5.5 - Vytvorenie a testovanie metódy `respawn(Enemy)` triedy `Arena`

Cieľ:

Učiteľ oboznámi žiakov s vytvorením a otestovaním metódy **`Arena.respawn(Enemy)`**, ktorá sa stará o opätovné objavenie nepriateľov v hre.

Koncepty na diskusiu:

Implementácia metódy, testovanie, herná mechanika.

Aktivita:

Učiteľ začne oboznámením žiakov s konceptom implementácie metódy a jej významom pri definovaní špecifického správania v triede. S dôrazom na praktickú aplikáciu vedie žiakov k vytvoreniu metódy **`respawn`** v triede **`Arena`**. Táto metóda, ktorá nevráti hodnotu, prijíma jeden parameter typu **`Enemy`**. Žiaci sú usmernení, aby v rámci tejto metódy nastavili polohu a natočenie nepriateľa tak, aby zodpovedali hodnotám pôvodne nastaveným v konštruktore. Učiteľ predvedie správnu syntax a štruktúru definovania tejto metódy, čím posilní kľúčové pojmy implementácie metódy a odovzdávania parametrov.

Žiaci následne otestujú svoju metódu, aby sa uistili, že funguje správne. Vytvoria inštanciu triedy **`Arena`** a **`Enemy`**, ale aplikáciu nespustia okamžite. Namiesto toho presunú myšou inštanciu triedy **`Enemy`** na nové miesto, potom vstúpia do kontextového menu inštancie triedy **`Arena`** a vyvolajú metódu **`respawn`**. Učiteľ vysvetlí proces zabezpečenia pozastavenia aplikácie a aktívneho poľa parametrov a usmerní žiakov, aby klikli pravým tlačidlom myši na inštanciu triedy **`Enemy`** a správne vyplnili pole parametrov. Žiaci pozorujú výraz zostavený v okne a potom kliknú na tlačidlo OK, aby videli výsledok použitia ich metódy **`respawn`**.

Prostredníctvom tejto aktivity žiaci získajú praktické skúsenosti s písaním a testovaním metód a pochopia, ako programovo manipulovať s hernými objektmi. Učiteľ dbá na to, aby žiaci pochopili každý krok, v prípade potreby im poskytuje pomoc a vysvetlenie, čím posilňuje ich porozumenie implementácii metód a hernej mechanike.

Commit: [43a221876b8acb4fd507175ec4c8f520121d1ab1](#)

7. Úloha 5.6 - Vytvorenie a testovanie metódy `hit(Enemy)` triedy `Orb` (10 minút)

Cieľ:

Učiteľ poučí žiakov o vytvorení a otestovaní metódy **Orb.hit(Enemy)**, ktorá definuje interakciu, keď nepriateľ zasiahne cieľové miesto.

Koncepty na diskusiu:

Interakcia metód, aktualizácia stavu objektu.

Aktivita:

Učiteľ na začiatku vysvetlí zmysel metódy **Orb.hit(Enemy)** a zdôrazní, že obsahuje logiku interakcie medzi cieľovým miestom a nepriateľom. Potom žiakov vedie k pridaniu tejto metódy do triedy **Orb**. Táto metóda, ktorá nevráti hodnotu, prijíma jeden parameter typu **Enemy**.

Pri testovaní tejto metódy žiaci postupujú krok za krokom podobne ako pri testovaní metódy **respawn**. Vytvoria inštanciu triedy **Orb** a inštanciu triedy **Enemy**. Bez spustenia aplikácie vyvolajú kontextové menu inštalácie triedy **Orb** a vyberú metódu **hit**. Učiteľ sa uistí, že žiaci pochopili, ako vyplniť pole parametrov kliknutím pravým tlačidlom myši na inštanciu triedy **Enemy**, keď je aplikácia pozastavená. Táto akcia vytvorí výraz volania metódy, ktorý potom žiaci vykonajú kliknutím na tlačidlo OK.

Učiteľ zdôrazní dôležitosť sledovania výrazu vytvoreného v okne na overenie volania metódy. Toto cvičenie pomáha žiakom pochopiť interakciu metód a proces aktualizácie stavov objektov v kontexte hry. Prostredníctvom tejto praktickej činnosti žiaci získajú skúsenosti s implementáciou a testovaním metód a upevnia si tak pochopenie interakcie metód a správania sa hry. Učiteľ poskytuje v prípade potreby podporu a vysvetlenie, čím zabezpečí, že žiaci úspešne dokončia úlohu a pochopia jej význam.

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

7.2. Greenfoot objekty a ich spolupráca: preskúmanie asociácií a pokročilých volaní metód

Tabuľka 15: Greenfoot objekty a ich spolupráca: preskúmanie asociácií a pokročilých volaní metód

<i>Názov</i>	Greenfoot objekty a ich spolupráca: preskúmanie asociácií a pokročilých volaní metód
<i>Výchovno-vzdelávacie ciele a ich opis</i>	Na konci lekcie by žiaci mali hlbšie porozumieť tomu, ako môžu objekty rôznych tried vytvárať asociácie a efektívne interagovať v prostredí Greenfoot . Žiaci by mali preukázať schopnosť zavolať metódu Orb.hit(Enemy) z triedy Enemy , čím demonštrujú svoju schopnosť zavádzať pokročilé volania metód a zjednodušovať komunikáciu medzi objektmi. Mali by byť schopní vysvetliť funkčnosť kľúčových metód z prostredia Greenfoot, ako napríklad Greenfoot.stop() a World.getWorldOfType(_cls_) , pochopiť ich úlohu pri riadení

	<p>vykonávania hry a efektívnej správe inštancií objektov a tiež by žiaci mali úspešne implementovať metódu Orb.hit(Enemy) v rámci svojich projektov a začleniť interakcie objektov a funkcionality metód na vytvorenie interaktívnej a dynamickej hernej mechaniky.</p> <p>Žiaci by mali aplikovať základné princípy objektovo orientovaného programovania, vrátane zapuzdrenia a volania metód, na vytvorenie sofistikovaných herných interakcií a funkcionalít. Mali by získať praktické poznatky o rôznych aspektoch vývoja hier vrátane vytvárania nepriateľov, riadenia stavu hry a vytvárania atraktívnych hráčskych dojmov prostredníctvom organizovanej interakcie objektov.</p>
<i>Cieľová skupina</i>	<p>Žiaci stredných škôl na kurze OOP4Fun. Základné znalosti programovania vrátane konceptov iterácie a výberu. Žiaci by mali byť oboznámení s programom Greenfoot.</p>
<i>Časové trvanie</i>	<ol style="list-style-type: none"> 1. Asociácia (10 minút) 2. Zavolanie metódy hit(Enemy) triedy Orb z triedy Enemy (15 minút) 3. Vysvetlenie kódu metód Greenfoot.stop() a World.getWorldOfType(_cls_) (15 minút) 4. Úloha 5.8 - Implementácia metódy hit(Enemy) triedy Orb (30 minút)
<i>Materiály a zdroje</i>	<ul style="list-style-type: none"> • Učebnica z projektu OOP4Fun. • Zdroje z projektu OOP4Fun. • Zdrojový kód projektu z Github/Gitlab. • Internetové zdroje.
<i>Popis</i>	<p>V tejto 70-minútovej lekcii sa žiaci stredných škôl podrobne oboznámia s asociáciami medzi objektmi a pokročilými volaniami metód v prostredí Greenfoot. Cieľom tejto lekcie je prehĺbiť porozumenie žiakov konceptom objektovo orientovaného programovania a zlepšiť ich schopnosť implementovať zložité interakcie pri vývoji hier.</p> <p>Na začiatku sa uskutoční 10-minútová diskusia o asociáciách medzi triedami so zameraním na to, ako môžu objekty v rámci softvérového systému vzájomne komunikovať a spolupracovať. Toto základné chápanie je základom pre skúmanie zložitejších interakcií.</p> <p>Následne sa žiaci zapoja do 15-minútovej úlohy zameranej na volanie metódy Orb.hit(Enemy) z triedy Enemy. Táto úloha kladie dôraz na praktické využitie volania metód a odovzdávania správ medzi objektmi.</p>

	<p>Ďalšia časť obsahuje podrobné vysvetlenie kódu metód Greenfoot.stop() a World.getWorldOfType(_cls_), čo zaberie 15 minút. Žiaci získajú prehľad o tom, ako tieto metódy fungujú v rámci prostredia Greenfoot a umožňujú presné ovládanie herných prvkov a správu sveta.</p> <p>Hlavná časť je venovaná 30-minútovej úlohe, v ktorej žiaci implementujú metódu Orb.hit(Enemy). Táto úloha je pre žiakov výzvou, aby využili svoje znalosti implementácie metód, odovzdávania parametrov a interakcie objektov na vytvorenie funkčnej hernej mechaniky v rámci svojich projektov.</p> <p>Na konci sa žiaci naučia lepšie chápať asociácie medzi objektmi, ovládať pokročilé volania metód, ako napríklad Orb.hit(Enemy) z triedy Enemy, a získajú prehľad o implementácii kľúčových metód prostredia Greenfoot. Tieto zručnosti ich pripraví na tvorbu interaktívnejších a dynamickejších hier, pričom budú môcť naplno využiť potenciál princípov objektovo orientovaného programovania pri vývoji hier.</p>
<i>Hodnotenie</i>	Gamifikácia predstavuje neformálne hodnotenie, ale zvýši záujem, vnútornú motiváciu a vzdelávacie výstupy celej skupiny.
<i>Šírenie výsledkov</i>	Na šírenie výsledkov medzi učiteľmi a spolužiakmi sa použije bežné nastavenie Learning management systému (napr. Moodle). Žiaci môžu pokračovať v diskusii na danú tému na fóre, ktoré im je poskytnuté prostredníctvom nástroja na Learning management.

7.2.1. Príručka učiteľa na prípravu vyučovacej hodiny

1. Asociácia

Cieľ:

Žiaci porozumejú pojmu asociácie medzi objektmi a pochopia, ako na seba môžu objekty rôznych tried vzájomne pôsobiť.

Koncepty na diskusiu:

Asociácie, interakcie objektov, vzťahy medzi triedami.

Aktivita:

Lekcia začína stručným prehľadom asociácií medzi triedami v objektovo orientovanom programovaní. Učiteľ zapojí žiakov do diskusie s cieľom objasniť, ako objekty navzájom komunikujú prostredníctvom asociácií, pričom na ilustráciu týchto pojmov použije

praktické príklady z prostredia Greenfoot. Žiaci sa hlbšie oboznámia a pochopia s pojmom asociácie. Budú vedieť, že asociácia definuje, ako triedy spolupracujú. Príkladom môže byť to, ako trieda **Enemy** ovplyvňuje triedu **Orb** v scenári hry.

Prostredníctvom podrobnej diskusie učiteľ rozoberie rôzne typy asociácií v rámci prostredia Greenfoot: jedna k jednej, jedna k viacerým a viaceré k viacerým. Napríklad asociácia jedna k jednej by mohla predstavovať to, ako je hráč prepojený so svojím avatarom postavy, asociácia jedna k viacerým by mohla predstavovať to, ako jeden svet obsahuje viacero inštancií aktérov, a asociácia viacero k viacerým by mohla zobrazovať to, ako rôzni nepriatelia interagujú s viacerými cieľovými miestami v rámci úrovne hry.

Pomocou UML diagramov tried určených pre Greenfoot učiteľ vizuálne znázorní tieto vzťahy a pomôže žiakom pochopiť, ako sú asociácie štruktúrované a implementované v ich herných projektoch. Diagram by napríklad mohol znázorňovať, ako je trieda **Enemy** spojená s viacerými inštanciami triedy **Orb**, čo naznačuje vzťah jedna k mnohým, kde každý nepriateľ ovplyvňuje niekoľko cieľových miest.

Žiaci sa aktívne podieľajú na rozpoznávaní a určovaní týchto asociácií v rámci svojich herných projektov. Skúmajú, ako objekty na základe týchto vzťahov interagujú, a diskutujú o dôsledkoch pre herné mechaniky a logiku. Učiteľ uvádza konkrétne príklady z kontextu ich vývoja hry, na ktorých ilustruje, ako inštancie triedy **Enemy** interagujú s inštanciami triedy **Orb** a ako sa tieto interakcie riadia asociáciami.

Na konci aktivity žiaci získajú pevnú predstavu o úlohe, ktorú asociácie zohrávajú pri navrhovaní a implementácii interaktívnych systémov v rámci prostredia Greenfoot. Dokážu identifikovať rôzne typy asociácií a použiť tieto znalosti na efektívne modelovanie a implementáciu komplexných interakcií medzi objektmi v prostredí Greenfoot. Toto praktické pochopenie posilňuje ich schopnosť navrhovať ucelené a interaktívne herné scenáre s využitím princípov objektovo orientovaného programovania.

2. [Zavolanie metódy `hit\(Enemy\)` triedy `Orb` z triedy `Enemy`](#)

Cieľ:

Učiteľ oboznámi žiakov s postupom volania metódy `Orb.hit(Enemy)` z triedy `Enemy`.

Koncepty na diskusiu:

Vyvolanie metódy, referencie na objekty.

Aktivita:

Žiaci sa zapoja do praktického učenia zmenou metódy `act()` triedy `Enemy`. Odstránia existujúci kód zodpovedný za nepotrebné správanie, ako je otáčanie pri dosiahnutí cieľového miesta alebo odrážanie sa od okrajov sveta. Namiesto toho implementujú funkcionality zavolania metódy `Orb.hit(Enemy)`, ak sa inštancia triedy `Enemy` stretne s inštanciou triedy `Orb`.

Učiteľ na praktických príkladoch a ukázkach ilustruje, ako správne pripraviť volanie metódy. Žiaci sa naučia používať referenciu na danú inštanciu (`this`) na spustenie metódy

hit() pre inštanciu triedy **Orb** pri stretnutí s inštanciou triedy **Enemy**. Preskúmajú priebeh riadenia v prostredí Greenfoot a pochopia, ako volanie metódy určuje cestu vykonania v rámci ich herného scenára.

Počas celej aktivity učiteľ poskytuje usmernenia na ladenie a testovanie implementácie, aby sa uistil, že volanie **Orb.hit(Enemy)** funguje tak, ako má. Žiaci pozorujú správanie v prostredí Greenfoot a overujú, či volanie metódy účinne spúšťa očakávané interakcie medzi inštanciami tried **Enemy** a **Orb**.

Na konci vyučovacej jednotky žiaci získajú zručnosti vo volaní metód a interakcii objektov v rámci programovania v prostredí Greenfoot. Pochopia, ako využívať referencie na objekty na vyvolanie metód v rôznych triedach, čím sa posilní ich pochopenie princípov objektovo orientovaného programovania v kontexte vývoja hier.

Commit: [63f9c96717d9d2587b60095e3b249b0158c8587b](#)

3. Vysvetlenie kódu metód `Greenfoot.stop()` a `World.getWorldOfType(_cls_)`

Cieľ:

Učiteľ vysvetlí funkčnosť metódy **Greenfoot.stop()** a metódy **World.getWorldOfType(_cls_)**.

Koncepty na diskusiu:

Metódy kontroly hry, správa sveta.

Aktivita:

V tejto vyučovacej jednotke sa žiaci oboznámia s dvoma kľúčovými metódami v rámci prostredia Greenfoot: **Greenfoot.stop()** a **World.getWorldOfType(_cls_)**. Učiteľ začne objasnením účelu a použitia každej z metód, pričom zdôrazní ich úlohy pri ovládaní hry a správe sveta.

Metóda **Greenfoot.stop()** je nevyhnutná na riadenie priebehu vykonávania scenára v prostredí Greenfoot. Po jej vyvolaní sa zastaví simulácia a zmrazia sa všetci aktéri a interakcie vo svete. Táto metóda je obzvlášť užitočná na implementáciu funkcie pozastavenia hry alebo na spustenie špecifických udalostí, ktoré si vyžadujú dočasné zastavenie postupu hry.

Avšak metóda **World.getWorldOfType(_cls_)** slúži na iný účel súvisiaci so správou sveta. Táto metóda umožňuje vývojárom získať inštancie svetov, ktoré sú konkrétneho typu triedy **_cls_**. Prechádza všetky aktívne svety v prostredí Greenfoot a vracia inštancie triedy sveta, ktoré zodpovedajú zadanému typu. Táto možnosť je výhodná, keď vývojári potrebujú dynamicky komunikovať so svetmi alebo s nimi manipulovať na základe atribútov ich tried.

Žiaci sa zúčastňujú na praktických ukážkach a príkladoch, aby lepšie pochopili tieto metódy. Učiteľ predvedie, ako možno metódu **Greenfoot.stop()** integrovať do herných scenárov s cieľom vytvoriť funkciu pozastavenia alebo spustiť špecifické udalosti v hre.

Žiaci pozorujú, ako pozastavenie hry ovplyvňuje správanie aktérov a interakcie v prostredí Greenfoot.

Podobne žiaci prostredníctvom praktických cvičení spoznávajú metódu `World.getWorldOfType(_cls_)`. Naučia sa, ako túto metódu používať na dynamické získavanie inštancií konkrétnych typov svetov. Učiteľ demonštruje scenáre, v ktorých je načítanie svetov konkrétneho typu triedy potrebné na implementáciu pokročilých herných mechaník alebo na správu viacerých súbežných herných prostredí v rámci prostredia Greenfoot.

Počas celej aktivity učiteľ podporuje diskusiu a poskytuje praktické príklady programovania na ilustráciu použitia týchto metód v reálnych scenároch vývoja hier. Žiaci sa aktívne podieľajú na experimentovaní s metódami v rámci vlastných projektov Greenfoot, čím si posilňujú porozumenie prostredníctvom priameho uplatňovania a skúmania.

Na konci vyučovacej jednotky žiaci získajú zručnosť v používaní funkcie `Greenfoot.stop()` na ovládanie hry a funkcie `World.getWorldOfType(_cls_)` na efektívnu správu sveta v prostredí Greenfoot. Získajú praktické zručnosti, ktoré zvýšia ich schopnosť implementovať komplexné herné správanie a efektívne spravovať stavy hry pomocou týchto základných metód.

4. Úloha 5.8 - Implementácia metódy `hit(Enemy)` triedy `Orb`

Cieľ:

Učiteľ vedie žiakov pri implementácii metódy `Orb.hit(Enemy)`.

Koncepty na diskusiu:

Implementácia metód, aktualizácia stavu objektov, herné mechaniky.

Aktivita:

Žiaci začnú implementovať metódu `Orb.hit(Enemy)`, ktorá je kľúčovým krokom pri definovaní interakcie medzi nepriateľom a cieľovým miestom v rámci ich herného scenára.

Metóda `Orb.hit(Enemy)` zohráva kľúčovú úlohu pri určovaní dôsledkov kontaktu nepriateľa s cieľovým miestom v hre. Tu je vysvetlený postup, ako môžu žiaci pristupovať k tejto metóde a implementovať ju. Najskôr je potrebné znížiť body zdravia inštancie triedy `Orb`. Po zavolaní metódy `Orb.hit(Enemy)` by sa mali znížiť body zdravia (hp) cieľového miesta. Táto akcia simuluje poškodenie, ktoré cieľové miesto utrpí pri kontakte s nepriateľom. Potom skontrolujte, či hp cieľového miesta klesli na nulu. Ak hp dosiahnu nulu alebo menej, hra by sa mala ukončiť. To sa zabezpečí vyvolaním funkcie `Greenfoot.stop()`, ktorá zastaví vykonávanie hry. To je prípad ukončenia hry. Iný prípad nastane, ak je hp cieľového miesta po stretnutí s nepriateľom stále nad nulou. Žiaci by mali zahrnúť logiku na opätovné objavenie nepriateľa späť v aréne, aby hra mohla pokračovať.

Počas aktivity učiteľ pomáha pri implementácii metódy **Orb.hit(Enemy)** tým, že poskytuje inštrukcie o štruktúre metódy, použití parametra typu **Enemy** a spôsobe aktualizácie **hp**. Žiaci spolupracujú, diskutujú a rozhodujú o konkrétnych herných mechanikách, ktoré chcú implementovať, napríklad o tom, aké poškodenie spôsobí cieľovému miestu každý typ nepriateľa a čo sa stane, keď **hp** cieľového miesta klesne pod hodnotu 1.

Učiteľ vyzve žiakov, aby svoje implementácie dôkladne otestovali a uistili sa, že sa metóda v rôznych herných scenároch správa podľa očakávania. Na konci aktivity žiaci získajú praktické skúsenosti s implementáciou logiky metódy na efektívne riadenie herných interakcií, čím sa posilní ich pochopenie implementácie metódy a hernej mechaniky v rámci prostredia Greenfoot.

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

7.3. Greenfoot objekty a ich spolupráca: Veže, strely a strategické interakcie

Tabuľka 16: Greenfoot objekty a ich spolupráca: Veže, strely a strategické interakcie

<i>Názov</i>	Greenfoot objekty a ich spolupráca: Veže, strely a strategické interakcie
<i>Výchovno-vzdelávacie ciele a ich opis</i>	Na konci lekcie získajú žiaci zručnosti pri vytváraní tried Bullet a Tower , čím si vytvoria základ pre vývoj strategických hier. Pochopia a implementujú realistický pohyb strely a určia akcie, keď sa inštanície triedy Bullet stretnú s inštanciami triedy Enemy alebo s okrajom arény. Žiaci navrhnu efektívnu streleckú mechaniku pre inštanície triedy Tower , pričom využijú odovzdávanie správ medzi inštanciami triedy Tower a ďalšími hernými prvkami na posilnenie hernej dynamiky. Strategicky rozmiestnia inštanície triedy Tower v rámci hernej arény a uplatnia objektovo orientované princípy, ako je zapuzdrenie a volanie metód, aby zabezpečili vytvorenie robustnej a dobre udržiavateľnej hernej mechaniky. Prostredníctvom spoločného riešenia problémov budú žiaci riešiť problémy pri navrhovaní a implementácii strategických interakcií medzi vežami, strelami a hernými prvkami, získajú praktické poznatky o princípoch herného dizajnu a zlepšia svoje celkové pochopenie hernej mechaniky v hre typu tower defense v prostredí Greenfoot.
<i>Cieľová skupina</i>	Žiaci stredných škôl na kurze OOP4Fun. Základné znalosti programovania vrátane konceptov iterácie a výberu. Žiaci by mali byť oboznámení s programom Greenfoot.

<p><i>Časové trvanie</i></p>	<ol style="list-style-type: none"> 1. Úloha 5.9 - Pridanie tried Bullet a Tower (10 minút) 2. Úloha 5.10 - Diskusia o tom, ako sa má pohybovať inštancia triedy Bullet a čo sa má stať, keď dosiahne inštanciu triedy Enemy alebo okraj arény. (10 minút) 3. Úloha 5.11 - Implementácia pohybu inštancie triedy Bullet (30 minút) 4. Úloha 5.12 - Diskusia o tom, ako inštancia triedy Tower vystrelí inštanciu triedy Bullet (15 minút) 5. Úloha 5.13 - Diskusia o tom, ako by mala inštancia triedy Tower komunikovať s príslušnými objektmi pomocou správ pri streľbe (15 minút) 6. Úloha 5.14 - Implementácia streľby inštancie triedy Tower (30 minút) 7. Úloha 5.15 - Veže v aréne (20 minút)
<p><i>Materiály a zdroje</i></p>	<ul style="list-style-type: none"> • Učebnica z projektu OOP4Fun. • Zdroje z projektu OOP4Fun. • Zdrojový kód projektu z Github/Gitlab. • Internetové zdroje.
<p><i>Popis</i></p>	<p>V tejto 130-minútovej lekcii sa žiaci stredných škôl oboznámia s dynamikou interakcií medzi vežami a strelami v prostredí Greenfoot. Scenár sa zameriava na rozvoj zručností žiakov pri navrhovaní a implementácii strategických herných prvkov s cieľom vytvoriť atraktívnu a interaktívnu hru.</p> <p>Vyučovanie sa začína 10-minútovou úlohou, v ktorej žiaci vytvoria triedy Bullet a Tower. Tento základný krok vytvára predpoklady pre pochopenie a implementáciu interakcií medzi týmito hernými prvkami.</p> <p>Potom nasleduje 10 minút diskusie o tom, ako by sa mali pohybovať inštancie triedy Bullet a aké akcie by sa mali vykonať, keď strela dosiahne inštanciu triedy Enemy alebo okraj arény. Táto diskusia je základom pre implementáciu presnej a dynamickej mechaniky pohybu.</p> <p>Žiaci potom venujú 30 minút implementácii pohybu inštancií triedy Bullet. Táto úloha je pre žiakov výzvou, aby použili svoje znalosti metódy <code>move()</code> v prostredí Greenfoot a obsluhu udalostí na simuláciu realistického správania sa strely v hernom prostredí.</p> <p>Pokračuje sa riešením úlohy v trvaní 15 minút, v ktorej sa diskutuje o tom, ako by mali inštancie triedy Tower strieľať inštancie triedy Bullet. Táto diskusia sa zaoberá logikou a podmienkami iniciovania streľby z</p>

	<p>veží pomocou triedy Bullet.</p> <p>Po nej nasleduje ďalších 15 minút venovaných diskusii o tom, ako by mali inštancie triedy Tower pri strelbe komunikovať s príslušnými objektmi pomocou správ. Tento segment zdôrazňuje dôležitosť komunikácie objektov a spúšťania udalostí pri vývoji hier.</p> <p>Žiaci potom stravia 30 minút implementáciou mechanizmu strelby inštancií triedy Tower. Táto úloha vyžaduje, aby žiaci začlenili logiku strelby do interakcie objektov a zabezpečili, aby veže efektívne zasahovali nepriateľov alebo iné herné prvky.</p> <p>Na záver tejto časti je 20 minútová úloha zameraná na správu a rozmiestňovanie veží v aréne. Pri tejto úlohe sa skúma umiestnenie, interakcia a strategické rozmiestnenie veží s cieľom optimalizovať dynamiku hry a vytvoriť skutočné výzvy pre hráčov.</p> <p>Na konci hodiny žiaci získajú praktické skúsenosti s navrhovaním a implementáciou strategických interakcií medzi vežami a strelami v prostredí Greenfoot. Získajú zručnosti v oblasti objektovo orientovaného programovania, spracovania udalostí a strategického návrhu hry, čo ich pripraví na tvorbu pútavých a dynamických hier.</p>
<i>Hodnotenie</i>	Gamifikácia predstavuje neformálne hodnotenie, ale zvýši záujem, vnútornú motiváciu a vzdelávacie výstupy celej skupiny.
<i>Šírenie výsledkov</i>	Na šírenie výsledkov medzi učiteľmi a spolužiakmi sa použije bežné nastavenie Learning management systému (napr. Moodle). Žiaci môžu pokračovať v diskusii na danú tému na fóre, ktoré im je poskytnuté prostredníctvom nástroja na Learning management.

7.3.1. Príručka učiteľa na prípravu vyučovacej hodiny

1. Úloha 5.9 - Pridanie tried **Bullet** a **Tower**

Cieľ:

Žiaci dokážu vytvoriť triedy **Bullet** a **Tower**.

Koncepty na diskusiu:

Vytvorenie triedy, roly triedy a počiatočné nastavenie.

Aktivita:

Lekcia začína prehľadom základných pojmov týkajúcich sa vytvárania objektov, pohybu a interakcie v prostredí Greenfoot. Učiteľ zapojí žiakov do diskusie s cieľom objasniť úlohy rôznych tried a ich interakcie v hre.

Žiaci vytvoria dve nové triedy v prostredí Greenfoot a pochopia účel každej z nich v kontexte hry. Naučia sa, ako tieto triedy nastaviť, čím sa pripravia na zložitejšie interakcie v neskorších lekciami.

Žiaci začnú vytvorením triedy **Bullet**. Táto trieda bude reprezentovať strely vystrelené vežami. V programe Greenfoot vyberú z ponuky triedy **Actor** (kliknutie pravým tlačidlom myši) vyberú položku „Nová podtrieda“ a novú triedu pomenujú **Bullet**.

Potom žiaci vytvoria triedu **Tower**. Táto trieda bude predstavovať veže, ktoré strieľajú strely na nepriateľov. Z ponuky triedy **Actor** opäť vyberú položku „Nová podtrieda“ a novú triedu pomenujú **Tower**.

Počas celej hodiny učiteľ vysvetľuje úlohy tried **Bullet** a **Tower** v hre. Trieda **Bullet** predstavuje strely, ktoré budú strieľať veže, zatiaľ čo trieda **Tower** predstavuje stacionárne objekty, ktoré môžu strieľať strely na nepriateľov. Učiteľ sa uistí, že žiaci rozumejú rozdielnym úlohám, ktoré každá trieda zohráva, a tomu, ako budú v hre vzájomne pôsobiť.

Na konci tejto aktivity by mali žiaci vytvoriť a nastaviť základné štruktúry tried **Bullet** a **Tower**, čím vytvoria základ pre podrobnejšiu implementáciu na ďalších hodinách.

Commit: [ece4df70042c8f60098e14ad2cee55514897d825](https://github.com/cece4df70042c8f60098e14ad2cee55514897d825)

2. Úloha 5.10 - Diskusia o tom, ako sa má pohybovať inštancia triedy **Bullet** a čo sa má stať, keď dosiahne inštanciu triedy **Enemy** alebo okraj arény.

Cieľ:

Učiteľ vedie diskusiu o očakávanom správaní strely pri jej pohybe v hre.

Koncepty na diskusiu:

Logika pohybu, detekcia kolízií.

Aktivita:

Žiaci diskutujú o logike pohybu strely a riešenia kolízií.

Žiaci začnú diskutovať o tom, ako by sa mala inštancia triedy **Bullet** pohybovať v hre. Mali by dospieť k záveru, že strela by sa mala pohybovať po priamke v smere, ktorým bola vystrelená, bez zmeny smeru. Rýchlosť strely by mala byť kontrolovateľná a konzistentná. Na realizáciu tohto cieľa môžu použiť metódy pohybu zabudované v prostredí Greenfoot. Je dôležité zdôrazniť úlohu konštruktora na inicializáciu hodnôt atribútov pri vytváraní inštancií objektov.

Učiteľ potom zameria diskusiu na to, čo by sa malo stať, keď strela dosiahne okraj sveta alebo sa stretne s nepriateľom. Žiaci by mali dospieť k záveru, že keď strela dosiahne okraj

sveta, mala by byť z hry odstránená. Diskutujú tiež o situácií, kedy by po strete s nepriateľom mala strela zmiznúť a nepriateľ by mal byť zranený alebo zničený.

Žiaci by potom mali navrhnúť pseudokód pre logiku pohybu a kolízie strely. Príkladom môže byť tento pseudokód: najprv sa strela pohybuje dopredu konštantnou rýchlosťou dopredu, pričom sa skontroluje, či strela dosiahla okraj sveta. Ak je to pravda, odstráni strelu zo sveta. Ak nie, skontroluje sa, či sa strela stretla s nepriateľom. Ak došlo ku stretu, znamená to, že strela zasiahla nepriateľa a je potrebné odstrániť strelu zo sveta a aplikovať poškodenie na nepriateľa alebo odstrániť nepriateľa.

Učiteľ predvedie, ako implementovať túto logiku pomocou metód v prostredí Greenfoot - `move(int)`, `isAtEdge()` a `getOneIntersectingObject(Class cls)`.

Učiteľ vyzve žiakov, aby svoje nápady dopracovali a premysleli ďalšie detaily, napríklad úpravu rýchlosti v závislosti od obtiažnosti hry alebo pridanie vizuálnych efektov, keď strela zasiahne nepriateľa. Táto aktivita pomáha žiakom pochopiť princípy detekcie pohybu a kolízií pri vývoji hier a pripravuje ich na ďalšiu implementáciu v ich projektoch.

3. Úloha 5.11 - Implementácia pohybu inštancie triedy `Bullet`

Cieľ:

Učiteľ pomáha žiakom implementovať logiku pohybu pre triedu `Bullet`.

Koncepty na diskusiu:

Pohyb aktora, hranice sveta.

Aktivita:

Žiaci napíšu kód na posúvanie strely dopredu a na jej odstránenie, keď dosiahne okraj arény. Žiaci začnú skúmaním konceptu pohybu aktéra a hraníc sveta, pričom sa zamerajú na to, ako sa tieto koncepty vzťahujú na triedu `Bullet`. Učiteľ pripomenie žiakom predchádzajúce úlohy a zdôrazní vedomosti, ktoré by mali použiť. Žiaci si pripomenú, ako pridať kód do metódy `act()` na spracovanie interakcií na okraji sveta. Taktiež si pripomenú, ako spracovať zmeny smeru, keď aktér vstúpi do konkrétnych buniek. Okrem toho si pamätajú, ako používať počítadlá a mechanizmy oneskorenia v metóde `act()` na riadený pohyb.

S týmito poznatkami žiaci pokračujú v implementácii logiky pohybu pre triedu `Bullet`. Začnú pridaním metódy `move(int)` do metódy `act()` triedy `Bullet`, aby sa strela mohla kontinuálne pohybovať dopredu.

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

4. Úloha 5.12 - Diskusia o tom, ako inštancia triedy `Tower` vystrelí inštanciu triedy `Bullet`

Cieľ:

Učiteľ diskutuje so žiakmi o logike strelby z veže.

Koncepty na diskusiu:

Vytvorenie objektu, volanie metódy.

Aktivita:

Žiaci si premyslia, ako veža vytvorí a vystrelí strely. Žiaci začnú diskutovať o celkovej logike potrebnej na to, aby veža mohla v hre vystreľovať strely. Zamerajú sa na kľúčové pojmy, ako je vytváranie objektu strely a vyvolávanie metód na jej vystrelenie. Učiteľ zdôrazní, že veža by nemala strieľať strely pri každom volaní metódy `act()`, podobne ako bol pohyb nepriateľa riešený pomocou mechanizmu oneskorenia.

Žiaci si formou diskusie sformulujú kroky potrebné na to, aby veža vystreľovala strely s prestávkami. Učiteľ ich vedie k tomu, aby zvážili použitie mechanizmu oneskorenia pri strelbe. Je potrebné vysvetliť, aká by mohla byť úloha konštruktora pre implementáciu tohto mechanizmu.

Učiteľ vysvetlí, že v triede `Tower` budú musieť vytvoriť nový atribút `shootDelay` a počítadlo `nextShootCounter`. Tieto atribúty budú riadiť frekvenciu strelby.

Učiteľ rozoberie príslušné kroky a metódy pre triedu `Tower`. Najprv sa v triede `Tower` zadefinujú atribúty `shootDelay` a `nextShootCounter`, pričom sa inicializuje počítadlo `nextShootCounter` na hodnotu 0 a atribút `shootDelay` sa nastaví na príslušnú hodnotu. Potom je potrebné upraviť metódu `act()` triedy `Tower`, aby bolo možné obsluhovať logiku strelby. Metóda by mala vytvoriť a vystreliť strelu až vtedy, keď `nextShootCounter` dosiahne hodnotu 0. Po vystrelení by sa `nextShootCounter` mal vynulovať na hodnotu `shootDelay`. Ak `nextShootCounter` nie je 0, mal by sa znížiť o hodnotu 1. Na konci by sa mala definovať samostatná metóda `fire()`, ktorá by sa starala o vytvorenie a vystrelenie strely. Táto metóda vytvorí inštanciu objektu `Bullet` a pridá ho do sveta.

Týmto postupom žiaci pochopia, ako implementovať logiku strelby oddelením príslušných krokov do metód triedy `Tower`. Učiteľ dbá na to, aby žiaci pochopili dôležitosť volania metód a vytvárania objektov, čím posilňuje ich pochopenie týchto pojmov v kontexte ich hry.

5. [Úloha 5.13 - Diskusia o tom, ako by mala inštancia triedy `Tower` komunikovať s príslušnými objektmi pomocou správ pri strelbe](#)

Cieľ:

Učiteľ vysvetlí, ako veža komunikuje so strelami a inými predmetmi pomocou správ.

Koncepty na diskusiu:

Odobzdávanie správ, volania metód.

Aktivita:

Žiaci diskutujú o mechanizme odobzdávania správ pri strelbe veže.

Na začiatku hodiny učiteľ vysvetlí koncept odobzdávania správ a jeho význam v objektovo orientovanom programovaní. Učiteľ by mal zdôrazniť, ako objekty v hre medzi sebou komunikujú pomocou metód, ktoré slúžia ako správy.

Na ilustráciu môže učiteľ použiť sekvenčný diagram UML, na ktorom popíše interakcie medzi inštanciami tried **Tower**, **Bullet** a **Arena**. Diagram vizuálne znázorňuje tok správ a volaní metód, čím pomáha žiakom pochopiť postupnosť interakcií.

Žiaci diskutujú o podrobnom postupe, ako by mala veža pri streľbe interagovať so strelami a inými predmetmi. Učiteľ vysvetlí, že keď sa veža rozhodne strieľať, pošle správu (volanie metódy) na vytvorenie inštancie triedy **Bullet** a pridá ju do arény. Táto interakcia sa spúšťa v rámci metódy **act()** triedy **Tower**. Učiteľ pomocou sekvenčného diagramu UML predvedie, ako veža posielala správu do prostredia Greenfoot na pridanie novej inštancie triedy **Bullet** do sveta. Veža potom odošle správu inštancii triedy **Bullet** a nastaví jej smer tak, aby zodpovedal aktuálnej rotácii veže. Tým sa zabezpečí, že sa strela bude pohybovať v určenom smere. keď je strela vytvorená a umiestnená, bude interagovať s inými objektmi v hre, ako sú nepriatelia alebo okraje sveta. Učiteľ vysvetlí, ako sa tieto interakcie spracúvajú pomocou metódy **act()** triedy **Bullet**, ktorá môže zahŕňať kontrolu kolízií a v prípade potreby odstránenie strely.

Učiteľ zdôrazňuje kooperatívnu povahu týchto interakcií a ukazuje, ako sa algoritmus rozdelí medzi spolupracujúce objekty. To pomáha žiakom oceniť modulárny dizajn a prehľadné formy komunikácie v rámci ich hry.

6. Úloha 5.14 - Implementácia streľby inštancie triedy **Tower**

Cieľ:

Učiteľ usmerňuje žiakov pri realizácii streleckého mechanizmu pre triedu **Tower**.

Koncepty na diskusiu:

Vytvorenie objektu, umiestnenie aktéra.

Aktivita:

Žiaci napíšu kód, ktorý umožní veži strieľať.

Na začiatku učiteľ vysvetlí celkový cieľ: implementovať mechanizmus streľby pre triedu **Tower**. Učiteľ potom rozdelí úlohu na zvládnuteľné kroky a vedie žiakov pri každom z nich.

Najprv si žiaci pripraví potrebné atribúty a konštruktory pre triedu **Tower**. Učiteľ vysvetlí, že veža potrebuje atribút na sledovanie toho, kedy môže strieľať. Potom žiaci vytvoria dve metódy v triede **Tower**: **boolean canShoot()** a **void Tower.fire()**. Spočiatku môžu tieto metódy vrátiť hodnotu **false**, resp. neurobiť nič, takže ich možno použiť v metóde **act()**. Metóda **act()** sa potom aktualizuje tak, aby používala tieto metódy. Učiteľ vysvetlí, že metóda **canShoot()** by mala vrátiť **true**, ak počítadlo **shootCounter** dosiahne hodnotu 0. Metóda **fire()** je implementovaná tak, aby vytvorila inštanciu triedy **Bullet** a správne ju umiestnila.

Učiteľ sa uistí, že žiaci rozumejú každej časti kódu, pričom kladie dôraz na vytvorenie strely, jej umiestnenie vo svete a nastavenie jej natočenia podľa veže.

Žiaci potom otestujú svoje riešenie spustením hry, umiestnením inštancie triedy **Tower** a overením, že strieľa strely vo vhodných intervaloch. Učiteľ vyzve žiakov, aby odstránili prípadné problémy a uistili sa, že strely sú vytvorené a pohybujú sa podľa očakávania.

Na konci aktivity majú žiaci implementovaný funkčný mechanizmus strelby pre vežu a upevnia si tak svoje znalosti o vytváraní objektov, umiestňovaní aktérov a volaní metód v prostredí Greenfoot.

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

7. Úloha 5.15 - Veže v aréne

Cieľ:

Učiteľ vedie žiakov k tomu, aby do hry zapojili veže a strely a vytvorili tak funkčnú arénu.

Koncepty na diskusiu:

Integrácia hier, testovanie.

Aktivita:

Žiaci umiestňujú veže do arény a testujú ich interakciu so strelami a nepriateľmi. Učiteľ na začiatku vysvetlí cieľ: začleniť veže do arény a zabezpečiť ich správnu interakciu so strelami a nepriateľmi. Súčasťou hodiny bude umiestňovanie veží do arény a testovanie ich správania v rámci herného prostredia.

Žiaci začnú umiestňovaním inštancií triedy **Tower** na rôzne pozície v aréne. Učiteľ vysvetlí, ako pridávať veže prostredníctvom programu Greenfoot a ako zabezpečiť, aby bola každá veža správne umiestnená.

Ďalej učiteľ predstaví koncept preťažovania konštruktorov. To je užitočné najmä pri vytváraní inštancií triedy **Tower** s rôznymi rotáciami. Učiteľ potom vedie žiakov aktualizáciou triedy **Tower** tak, aby obsahovala preťažený konštruktor, ktorý prijíma celočíselný parameter pre rotáciu. To umožňuje väčšiu kontrolu nad umiestnením a orientáciou veží v aréne.

Commit: [bfb6a271f490c341c760e654b3f86a87111c54cb](https://github.com/bfb6a271f490c341c760e654b3f86a87111c54cb)

7.4. Greenfoot objekty a ich spolupráca: strely, nepriatelia a dynamika hry

Tabuľka 17: Greenfoot objekty a ich spolupráca: strely, nepriatelia a dynamika hry

<i>Názov</i>	Greenfoot objekty a ich spolupráca: strely, nepriatelia a dynamika hry
<i>Výchovno-vzdelávacie</i>	Na konci lekcie získajú žiaci zručnosti v navrhovaní a implementácii interaktívnej hernej dynamiky pomocou tried Bullet a Enemy v prostredí Greenfoot. Pochopia, ako uľahčiť interakciu objektov prostredníctvom

<p><i>ciele a ich opis</i></p>	<p>odovzdávania správ, čo umožní efektívnu komunikáciu medzi hernými prvkami. Žiaci prejavia schopnosť implementovať presné mechanizmy detekcie kolízií a reakcie, konkrétne s podrobným popisom interakcie inštancií triedy Bullet s inštanciami triedy Enemy. Získajú praktické poznatky o základných metódach prostredia Greenfoot, ako sú Greenfoot.showText(String, int, int), Greenfoot.getRandomNumber(int) a World.act(), a využijú ich na zlepšenie prezentácie hry, zavedenie náhodnosti a správu aktualizácií stavu hry. Okrem toho žiaci zvládnu implementáciu mechaniky vytvárania nepriateľov a podmienok ukončenia hry, čím sa zabezpečí dynamický zážitok z hry. Prostredníctvom prehodnotenia asociácií objektov si žiaci upevnia svoje chápanie toho, ako objekty spolupracujú pri vytváraní atraktívnej hernej dynamiky, čím sa pripravia na uplatnenie týchto zručností v budúcich projektoch vývoja hier.</p>
<p><i>Cieľová skupina</i></p>	<p>Žiaci stredných škôl na kurze OOP4Fun. Základné znalosti programovania vrátane konceptov iterácie a výberu. Žiaci by mali byť oboznámení s programom Greenfoot.</p>
<p><i>Časové trvanie</i></p>	<ol style="list-style-type: none"> 1. Úloha 5.16 - Diskusia o tom, ako by mala inštancia triedy Bullet komunikovať s príslušnými objektmi pomocou správ (15 minút) 2. Úloha 5.17 - Implementácia zásahu inštancie triedy Enemy inštanciou triedy Bullet (30 minút) 3. Vysvetlenie kódu metód <code>Greenfoot.showText(String, int, int)</code>, <code>Greenfoot.getRandomNumber(int)</code> a <code>World.act()</code> (15 minút) 4. Úloha 5.18 - Vznik nepriateľov a koniec hry (30 minút) 5. Prehľad o asociáciách (20 minút)
<p><i>Materiály a zdroje</i></p>	<ul style="list-style-type: none"> • Učebnica z projektu OOP4Fun. • Zdroje z projektu OOP4Fun. • Zdrojový kód projektu z Github/Gitlab. • Internetové zdroje.
<p><i>Popis</i></p>	<p>V tejto lekcii, ktorá trvá 110 minút, sa žiaci stredných škôl oboznámia so špecifickými princípmi dynamiky hry zahŕňajúcimi strely a nepriateľov v prostredí Greenfoot. Hodina sa zameriava na rozvoj zručností žiakov pri vytváraní interaktívnej a dynamickej hry prostredníctvom efektívnych interakcií objektov a herných mechaník.</p> <p>Hodina začína 15 minútovou diskusiou o tom, ako by mali inštancie triedy Bullet interagovať s príslušnými hernými objektmi pomocou správ. Táto</p>

	<p>diskusia predstavuje východisko pre pochopenie toho, ako objekty komunikujú a spolupracujú s cieľom dosiahnuť špecifické herné správanie.</p> <p>Následne sa žiaci budú 30 minút venovať implementácii funkcie, pri ktorej inštancie triedy Bullet úspešne zasiahnu inštancie triedy Enemy. Táto úloha je pre žiakov výzvou, aby uplatnili svoje znalosti o detekcii kolízií objektov a spracovaní udalostí na vytvorenie pôsobivých interakcií v hre.</p> <p>Ďalší segment obsahuje 15 minútové vysvetlenie základných metód prostredia Greenfoot: Greenfoot.showText(String, int, int), Greenfoot.getRandomNumber(int) a World.act(). Žiaci získajú prehľad o tom, ako tieto metódy prispievajú k zobrazovaniu textu, generovaniu náhodných čísel pre herné mechaniky a riadeniu cyklu aktualizácie herného sveta.</p> <p>Žiaci potom stravia 30 minút riešením úloh týkajúcich sa vytvárania nepriateľov a podmienok na konci hry. Patrí sem návrh a implementácia mechanizmov na vytváranie nepriateľov vo vhodných intervaloch a určenie podmienok ukončenia hry na základe akcií hráča alebo cieľov hry.</p> <p>Nasleduje 20 minútová kontrola, ktorá sa zameriava na upevnenie pochopenia asociácií objektov a ich úloh pri realizácii hernej dynamiky. Žiaci si zopakujú a zdokonalia svoje chápanie interakcie a spolupráce objektov v prostredí Greenfoot s cieľom dosiahnuť požadované herné efekty.</p> <p>Na záver lekcie žiaci získajú hlbšie pochopenie toho, ako vytvárať interaktívnu a pútavú hernú dynamiku zahŕňajúcu strely, nepriateľov a strategické herné mechaniky v prostredí Greenfoot. Budú vybavení praktickými zručnosťami pri implementácii interakcií objektov, správe herných stavov a zlepšovaní zážitkov hráčov prostredníctvom štruktúrovaných princípov herného dizajnu.</p>
<p><i>Hodnotenie</i></p>	<p>Gamifikácia predstavuje neformálne hodnotenie, ale zvýši záujem, vnútornú motiváciu a vzdelávacie výstupy celej skupiny.</p>
<p><i>Šírenie výsledkov</i></p>	<p>Na šírenie výsledkov medzi učiteľmi a spolužiakmi sa použije bežné nastavenie Learning management systému (napr. Moodle). Žiaci môžu pokračovať v diskusii na danú tému na fóre, ktoré im je poskytnuté prostredníctvom nástroja na Learning management.</p>

7.4.1. Príručka učiteľa na prípravu vyučovacej hodiny

1. Úloha 5.16 - Diskusia o tom, ako by mala inštancia triedy `Bullet` komunikovať s príslušnými objektmi pomocou správ

Cieľ:

Žiaci vedia kriticky a tvorivo diskutovať o tom, ako by mali strely interagovať s inými objektmi, najmä s nepriateľmi, prostredníctvom odovzdávania správ.

Koncepty na diskusiu:

Odovzdávanie správ, detekcia kolízií a interakcia objektov.

Aktivita:

Na začiatku sa preberajú interakcie objektov v prostredí Greenfoot so zameraním na to, ako inštancie rôznych tried komunikujú a navzájom sa ovplyvňujú. Učiteľ zapojí žiakov do diskusie s cieľom posilniť tieto pojmy a ich praktické využitie pri vývoji hier.

Žiaci diskutujú o logike interakcií so strelami a o správach, ktoré musia posilať. Učiteľ začne vysvetľovaním dôležitosti odovzdávania správ v objektovo orientovanom programovaní. Diskusia sa potom zameria na to, ako inštancie triedy `Bullet` interagujú s inými objektmi, ako sú nepriatelia a aréna, najmä keď strela zasiahne nepriateľa.

Žiaci sú vyzvaní, aby sa v rámci diskusie podelili o svoje nápady týkajúce sa logiky interakcie so strelami. Uvažujú o otázkach, ako napríklad: Čo by sa malo stať, keď strela zasiahne nepriateľa? Ako by mala strela oznámiť túto udalosť ostatným objektom? Aké správy je potrebné odovzdať na správne spracovanie interakcie?

Učiteľ predstaví koncept detekcie kolízie a vysvetlí, ako musí hra zistiť, kedy strela zasiahne nepriateľa. Diskutuje aj o následných akciách, ako je zníženie zdravia nepriateľa alebo odstránenie nepriateľa z arény.

Na vizualizáciu týchto interakcií učiteľ použije sekvenčný diagram UML. Diagram znázorňuje správy vymieňané medzi triedami `Bullet`, `Enemy` a `Arena` počas interakcie.

2. Úloha 5.17 - Implementácia zásahu inštancie triedy `Enemy` inštanciou triedy `Bullet`

Cieľ:

Učiteľ pomáha žiakom pri realizácii interakcie medzi strelou a nepriateľom.

Koncepty na diskusiu:

Detekcia kolízií, volanie metód a zmeny stavu objektu.

Aktivita:

Žiaci napíšu kód na spracovanie kolízie medzi strelou a nepriateľom, vrátane účinkov kolízie. Učiteľ na začiatku vysvetlí koncepty detekcie kolízie a volania metód v prostredí Greenfoot a zdôrazní, že sú kľúčové pre spracovanie interakcií medzi hernými objektmi.

Žiaci budú krok za krokom vedení k implementácii logiky kolízie medzi strelami a nepriateľmi.

Žiaci si najprv pripravlia potrebné atribúty a metódy v triedach **Bullet** a **Enemy**. Žiaci napíšu kód v triede **Bullet** na detekciu kolízie s inštanciou triedy **Enemy** a zavolajú metódu **hit(Bullet)** z triedy **Enemy**. Žiaci potom implementujú metódu **hit(Bullet)** v triede **Enemy** na spracovanie účinkov kolízie, ako je zníženie zdravia nepriateľa alebo jeho odstránenie z hry. Žiaci otestujú svoju implementáciu, aby sa uistili, že interakcia medzi strelou a nepriateľom funguje tak, ako má.

Na konci hodiny budú mať žiaci funkčný mechanizmus detekcie kolízie medzi strelami a nepriateľmi s príslušnými volaniami metód a zmenami stavu objektu. Toto cvičenie posilní ich chápanie detekcie kolízií, volania metód a praktického využitia týchto konceptov pri vývoji hier.

Commit: [dcfe31bc006b7f3dcd8b8b759cc1be901c32913c](https://github.com/006b7f3dcd8b8b759cc1be901c32913c)

3. [Vysvetlenie kódu metód `Greenfoot.showText\(String, int, int\)`, `Greenfoot.getRandomNumber\(int\)` a `World.act\(\)`](#)

Cieľ:

Učiteľ vysvetlí použitie konkrétnych metód prostredia Greenfoot, ktoré budú použité v hre.

Koncepty na diskusiu:

Zobrazenie textu, generovanie náhodných čísel a metóda **act()**.

Aktivita:

Žiaci sa naučia zobrazovať text na obrazovke, generovať náhodné čísla a implementovať logiku hry v metóde **act()**.

Učiteľ začne predstavením metód **Greenfoot.showText(String, int, int)**, **Greenfoot.getRandomNumber(int)** a **World.act()**. Rozoberá účel týchto metód a zdôrazňuje ich význam pri vývoji hier na zobrazovanie informácií, vytváranie náhodnosti a definovanie správania.

Metóda **Greenfoot.showText(String, int, int)** sa používa na zobrazenie textu na obrazovke na zadaných súradniciach. Učiteľ vysvetlí, že táto metóda je užitočná na zobrazovanie informácií o hre, ako sú skóre, zdravie alebo pokyny priamo na obrazovke hry.

Metóda **Greenfoot.getRandomNumber(int)** vygeneruje náhodné číslo medzi 0 (vrátane) a zadanou hodnotou (táto sa nevygeneruje, predstavuje hornú hranicu). Učiteľ diskutuje o tom, ako možno túto metódu použiť na zavedenie náhodnosti do hry, napríklad na vytváranie nepriateľov na náhodných miestach alebo na generovanie náhodných vzorov pohybu.

Metóda `act()` triedy `World` je opakovane volaná prostredím `Greenfoot` na vykonanie hlavnej logiky hry. Učiteľ zdôrazňuje, že metóda `act()` je miestom, kde sú umiestnené hlavné herné akcie a logika, čo umožňuje priebežné aktualizácie a interakcie v hre.

4. Úloha 5.18 - Vznik nepriateľov a koniec hry

Cieľ:

Učiteľ pomáha žiakom implementovať vytváranie nepriateľov a podmienky na konci hry.

Koncepty na diskusiu:

Tvorba objektov, herná slučka a podmienky ukončenia hry.

Aktivita:

Žiaci napíšu kód, ktorý pravidelne vytvára nepriateľov a definuje podmienky, ktoré vyvolajú koniec hry. Učiteľ začne vysvetľovaním dôležitosť vytvárania nepriateľov v pravidelných intervaloch a definovaním podmienok konca hry, keď sú všetci nepriatelia porazení. Rozoberú sa pojmy vytvárania objektov, herných slučiek a podmienok konca hry, vďaka čomu žiaci jasne pochopia, čo je potrebné implementovať.

Na pravidelné volanie procesu vytvárania nepriateľov sa použije metóda `Arena.act()`. Mal by sa zaviesť mechanizmus oneskorenia na kontrolu intervalu medzi vytváraním nepriateľov. Vytvorte metódu `spawn()` v triede `Arena`, ktorá bude spracovávať samotný proces vytvárania nepriateľov. Táto metóda vytvorí inštanciu triedy `Enemy`, priradí vlastnosti nepriateľa (napr. pozíciu, atribúty) a pridá inštanciu nepriateľa do arény.

Žiaci definujú a implementujú podmienky konca hry. Keď počet nepriateľov klesne na nulu, hra sa skončí a hráč vyhrá. Na tento účel by žiaci mali v triede `Arena` udržiavať atribút, ktorý bude sledovať počet vytvorených nepriateľov. Tento atribút zvýšte pri každom vytvorení nepriateľa a znížte ho, keď je nepriateľ zabitý. Metóda `Arena.kill(Enemy)` by mala byť upravená tak, aby kontrolovala, či sú všetci nepriatelia porazení. Ak sú všetci nepriatelia zabití, zastavte hru pomocou `Greenfoot.stop()` a zobrazte správu o víťazstve. Volanie `Greenfoot.stop()` by preto malo byť posledným príkazom v metóde.

Na konci otestujte mechanizmus vytvárania nepriateľov pozorovaním pravidelného vytvárania nepriateľov v aréne. Uistite sa, že oneskorenie medzi ich vytvoreniami funguje správne. Overte stav konca hry simuláciou porážky všetkých nepriateľov a skontrolujte, či sa hra zastaví so zobrazenou správou o víťazstve.

Na konci tejto lekcie budú mať žiaci implementovaný funkčný systém vytvárania nepriateľov a definované jasné podmienky konca hry, čím si upevnia svoje znalosti o herných slučkách, správe objektov a výsledkoch hry založených na podmienkach.

Commit: [d48341a095561500af6032d5c8f56e201060f9a4](https://github.com/48341a095561500af6032d5c8f56e201060f9a4)

5. Prehľad o asociáciách

Cieľ:

Učiteľ zopakuje koncept asociácií medzi triedami a zdôrazní, ako objekty v prostredí Greenfoot interagujú a komunikujú.

Koncepty na diskusiu:

Asociácie, komunikácia objektov a odovzdávanie správ.

Aktivita:

Žiaci diskutujú a overujú si svoje chápanie asociácií, pričom vytvárajú súvislosti medzi rôznymi predmetmi a ich vzájomným pôsobením.

Učiteľ začne zopakovaním kľúčových pojmov týkajúcich sa asociácií medzi triedami. Patrí sem aj to, ako na seba objekty vzájomne pôsobia, komunikujú a odovzdávajú si správy. Na uľahčenie tohto postupu môže učiteľ vychádzať z príkladov a úloh preberaných na predchádzajúcich hodinách, čím pomôže žiakom upevniť si vedomosti a pochopiť, ako sa tieto pojmy uplatňujú v prostredí Greenfoot.

Na vizuálne znázornenie interakcií medzi rôznymi objektmi použite sekvenčné diagramy UML. Ukážte napríklad postupnosť správ, keď strela zasiahne nepriateľa, a ako aréna spracúva vytváranie a odstraňovanie nepriateľov.

Na konci tejto lekcie budú mať žiaci upevnené vedomosti o asociáciách a interakciách objektov v prostredí Greenfoot. Budú schopní jasne vyjadriť, ako rôzne objekty v hre komunikujú a spolupracujú, a aplikovať tieto koncepty na svoje vlastné projekty vývoja hier.

8. Dedičnosť

V rámci tematického celku Premenné a výrazy boli vytvorené štyri metodické materiály.

8.1. Úvod do dedičnosti v prostredí Greenfoot

Tabuľka 18: Úvod do dedičnosti v prostredí Greenfoot

<i>Názov</i>	Úvod do dedičnosti v prostredí Greenfoot
<i>Výchovno-vzdelávacie ciele a ich opis</i>	Na konci tejto lekcie budú žiaci schopní porozumieť konceptom dedičnosti . Pochopenie skúmaných pojmov sa bude rozoberať v kontexte vývoja hier, čo podporí kreativitu, tímovú prácu a nadšený prístup k programovaniu s využitím nástroja Greenfoot.
<i>Cieľová skupina</i>	Žiaci stredných škôl na kurze OOP4Fun. Základné znalosti programovania a základné znalosti objektovo orientovaného programovania. Žiaci by mali byť oboznámení s programom Greenfoot.
<i>Časové trvanie</i>	<ol style="list-style-type: none">1. Základné pojmy dedičnosti (15 minút)2. Hierarchia tried a dedičnosť (15 minút)3. Úlohy 6.1 - Identifikácia spoločných vlastností tried Orb a Direction (15 minút) a 6.2 - Pridanie triedy PassiveActor ako predka tried Orb a Direction (15 minút)4. Úvod do abstraktných tried (5 minút)5. Úloha 6.3 - Zavedenie abstraktnosti pre triedu PassiveActor (10 minút)
<i>Materiály a zdroje</i>	<ul style="list-style-type: none">• Učebnica z projektu OOP4Fun.• Zdroje z projektu OOP4Fun.• Zdrojový kód projektu z Github/Gitlab.• Internetové zdroje.
<i>Popis</i>	V tejto 75-minútovej lekcii sa žiaci stredných škôl oboznámia s princípmi objektovo orientovaného programovania súvisiacimi s dedičnosťou cez optiku vývoja hier pomocou nástroja Greenfoot. Vyučovanie sa začína 15-minútovým úvodom učiteľa o základných konceptoch dedičnosti, pričom sa vytvorí kontext súvisiaci s predchádzajúcimi stretnutiami a budúcim vývojom hry.

	<p>Potom nasleduje 15-minútový úsek, počas ktorého žiaci a učitelia diskutujú o hierarchii tried v ich hre. S cieľom vysvetliť pojmy súvisiace s dedičnosťou sa pozorujú triedy Orb a Direction. Počas ďalšej 15-minútovej úlohy sa skúma identifikácia spoločných vlastností týchto tried. Pozoruje sa, že tieto triedy počas svojho života nekonajú, iba reagujú na správy. Následne sa identifikuje spoločná metóda pre konanie, metóda act(). Táto metóda bude v oboch triedach definovaná s prázdny telom. Na základe identifikovaných spoločných vlastností bola v nasledujúcich 15 minútach implementovaná nová trieda PassiveActor.</p> <p>V ďalšom 5-minútovom úseku sa predstavia abstraktné triedy. Abstraktné triedy slúžia ako vzory pre iné triedy a nemožno ich inštanciovať. Sú však nevyhnutné pri navrhovaní hierarchií tried. Vzhľadom na to, že trieda PassiveActor je plánom pre konanie, v nasledujúcom 10-minútovom úseku je definovaná ako abstraktná trieda. Okrem toho je trieda PassiveActor stanovená ako predok tried Orb a Direction, čím sa triedy Orb a Direction stávajú jej potomkami. Keďže metóda act() je už definovaná v triede PassiveActor, je z tried Orb a Direction odstránená.</p> <p>Výsledkom je, že na konci hodiny sa žiaci oboznámia s novými pojmami súvisiacimi s dedičnosťou.</p>
<i>Hodnotenie</i>	<p>Gamifikácia predstavuje neformálne hodnotenie, ale zvýši záujem, vnútornú motiváciu a výsledky vzdelávania celej skupiny.</p> <p>Vzhľadom na dôležitosť konceptov dedičnosti otvára štruktúra projektu možnosti pre ďalšiu diskusiu a modifikáciu. V tejto súvislosti možno uvažovať o ďalších triedach a ich hierarchii a zaviesť ďalšie triedy, metódy a atribúty. Na druhej strane môže učiteľ túto tému upraviť tak, aby ukázal výhody dedičnosti a s ňou spojenú univerzálnosť len na tu navrhnutých hierarchiách.</p>
<i>Šírenie výsledkov</i>	<p>Na šírenie výsledkov medzi učiteľmi a spolužiakmi sa použije bežné nastavenie Learning management systému (napr. Moodle). Žiaci môžu pokračovať v diskusii na danú tému na fóre, ktoré im je poskytnuté prostredníctvom nástroja na Learning management.</p>

8.1.1. Príručka učiteľa na prípravu vyučovacej hodiny

1. Základné pojmy dedičnosti

Cieľ:

Žiaci sa oboznámia s konceptom dedičstva na príkladoch z reálneho života a diskusia o jeho výhodách.

Pojmy na diskusiu:

Dedičnosť, príklady dedičnosti v reálnom svete.

Aktivita:

V úvodnej časti je uvedený kontext súvisiaci s predchádzajúcimi lekciami. Učiteľ predstaví koncept dedičnosti. Učiteľ by mal tento pojem žiakom priblížiť pomocou príkladov z reálneho života (napr. ak sa uvažuje o vzťahu rodič - dieťa, deti dedia vlastnosti po svojich rodičoch, ako je typ vlasov, farba očí atď.) Mal by diskutovať o výhodách dedičnosti. Tieto pojmy sa posudzujú v kontexte prostredia Greenfoot a programovacieho jazyka Java.

2. Hierarchia tried a dedičnosť

Cieľ:

Predstavenie hierarchie tried v kontexte dedičnosti prostredníctvom vysvetlenia tried predkov a potomkov, diskusia o reálnych príkladoch a dedičnosti vlastností, zdôraznenie výhod hierarchie tried.

Pojmy na diskusiu:

Dedičnosť, hierarchia tried, príklady hierarchie tried v reálnom svete, výhody dedičnosti a hierarchie tried.

Aktivita:

Učiteľ predstaví hierarchiu tried v kontexte konceptu dedičnosti. Učiteľ predstaví triedy predkov (známe aj ako: nadtrieda, rodičovská trieda) a triedy potomkov (známe aj ako: podtriedy, detské triedy):

- v tomto kontexte možno diskutovať o predtým preskúmaných reálnych triedach,
- v tomto kontexte by sa malo diskutovať o tom, že podtriedy môžu dediť vlastnosti (t. j. atribúty a metódy) od rodičovskej triedy,
- malo by sa však diskutovať o tom, že podtriedy môžu obsahovať ďalšie vlastnosti, ktoré nie sú dostupné v rodičovskej triede.

Mali by sa prediskutovať výhody hierarchie tried v kontexte koncepcie dedičnosti. Malo by sa vysvetliť, že v programovacom jazyku Java môže mať každá trieda viac podtried, ale len jednu rodičovskú triedu. Možno diskutovať o úlohe triedy **Object** v kontexte hierarchie tried a dedičnosti.

3. Úlohy 6.1 - Identifikácia spoločných vlastností tried Orb a Direction a 6.2 - Pridanie triedy PassiveActor ako predka tried Orb a Direction

Cieľ:

Identifikácia spoločných vlastností v triedach hry, nájdenie triedy predka a implementácia novej triedy v hierarchii tried.

Pojmy na diskusiu:

Dedičnosť, hierarchia tried, implementácia dedičnosti a hierarchie tried pri vývoji hier.

Aktivita:

V kontexte vývoja hier sa uvažuje o triedach **Orb** a **Direction**. Je potrebné poznamenať, že tieto triedy reagujú na správy. Preto by sa mala identifikovať spoločná metóda pre konanie, metóda **act()**. Na základe identifikovaných spoločných vlastností by sa mala implementovať nová trieda **PassiveActor** obsahujúca metódu **act()**:

- Tieto triedy (**PassiveActor**, **Orb** a **Direction**) by sa mali použiť na reprezentáciu hierarchie tried v kontexte dedičnosti.
- Učiteľ môže vizuálne znázorniť hierarchiu tried pomocou diagramu hierarchie.
- Učiteľ upozorní žiakov, čo sa zmenilo v prostredí Greenfoot, keď sa v triede **Actor** nahradil triedou **PassiveActor**

Commit: [afe617814c07a5d885ed06479bf71deda8725f19](https://github.com/af617814c07a5d885ed06479bf71deda8725f19)

4. Úvod do abstraktných tried

Cieľ:

Predstavenie konceptu abstraktných tried, diskusia o ich úlohe ako vzorov pri navrhovaní hierarchií tried a preskúmanie reálnych príkladov na ilustráciu ich použitia a špecializácie na podtriedy.

Pojmy na diskusiu:

Dedičnosť, hierarchia tried, abstraktné triedy, reálne príklady abstraktných tried v kontexte dedičnosti a hierarchie tried.

Aktivita:

Učiteľ zavedie pojem abstraktná trieda. Je potrebné diskutovať o tom, že abstraktné triedy slúžia ako vzory pre iné triedy a nemožno ich inštanciovateľ. Sú však nevyhnutné pri navrhovaní hierarchií tried. Učiteľ a žiaci môžu diskutovať o príkladoch z reálneho sveta týkajúcich sa abstraktných tried a podtried (napr. trieda Počítač so základnými vlastnosťami môže byť definovaná ako abstraktná trieda a môže byť špecializovaná na konzolu, stolný počítač, prenosný počítač a mobilný telefón, pričom každá z nich má špecifickú sadu vlastností atď.) Ďalším príkladom môžu byť geometrické tvary. Obdĺžnik alebo trojuholník možno zdediť z abstraktnej triedy geometrický tvar. Pri výpočte obvodu a plochy všeobecného geometrického tvaru nemáme k dispozícii presný vzorec. Ale pre

obdĺžnik a trojuholník máme presný vzorec. Štvorec možno zdediť z obdĺžnika. Žiaci by mali diskutovať o ďalších príkladoch geometrických tvarov a telies.

5. Úloha 6.3 - Zavedenie abstraktnosti pre triedu `PassiveActor`

Cieľ:

Diskusia o úlohe triedy `PassiveActor` a implementácia tejto triedy ako abstraktnej.

Pojmy na diskusiu:

Dedičnosť, hierarchia tried, abstraktne triedy, implementácia abstraktnej triedy pri vývoji hry.

Aktivita:

Koncept abstraktnej triedy je zohľadnená v prostredí Greenfoot a programovacím jazyku Java. V kontexte vývoja hier je trieda `PassiveActor` vzorom pre hranie. Preto je definovaná ako abstraktná trieda a ustanovená ako predok tried `Orb` a `Direction`, čím sa triedy `Orb` a `Direction` stávajú jej potomkami. Keďže metóda `act()` je už definovaná v triede `PassiveActor`, mala by sa z tried `Orb` a `Direction` odstrániť.

Commit: [f7a5702cae29bf21c9c88620d01ef64e4127c21c](https://github.com/f7a5702cae29bf21c9c88620d01ef64e4127c21c)

8.2. Koncepty dedičnosti v prostredí Greenfoot (časť 1)

Tabuľka 19: Koncepty dedičnosti v prostredí Greenfoot

<i>Názov</i>	Koncepty dedičnosti v prostredí Greenfoot
<i>Výchovno-vzdelávacie ciele a ich opis</i>	Na konci tejto časti budú žiaci rozumieť ďalším konceptom dedičnosti . Preskúmané pojmy sa budú rozoberať v kontexte vývoja hier, čím sa podporí tvorivé myslenie, tímová práca a nadšený prístup k programovaniu v prostredí Greenfoot.
<i>Cieľová skupina</i>	Žiaci stredných škôl, ktorí sa zúčastňujú kurzu OOP4Fun. Základné znalosti programovania a základné znalosti objektovo orientovaného programovania. Žiaci by mali byť oboznámení s programom Greenfoot.
<i>Časové trvanie</i>	<ol style="list-style-type: none"> 1. Úloha 6.4 - Identifikácia spoločných vlastností tried <code>Bullet</code> a <code>Enemy</code> (15 minút) 2. Úloha 6.5 - Pridanie abstraktnej triedy <code>MovingActor</code> ako predka tried <code>Bullet</code> a <code>Enemy</code> (15 minút) 3. Úloha 6.6. - Identifikácia atribútov tried <code>Bullet</code> a <code>Enemy</code> potrebných na pohyb (15 minút)

	<p>4. Úvod do kľúčového slova <code>super</code> v kontexte dedičnosti (20 minút)</p> <p>5. Úloha 6.7. - Presun kódu zodpovedného za pohyb do triedy <code>MovingActor</code> (30 minút)</p>
<p><i>Materiály a zdroje</i></p>	<ul style="list-style-type: none"> • Učebnica z projektu OOP4Fun. • Zdroje z projektu OOP4Fun. • Zdrojový kód projektu z Github/Gitlab. • Internetové zdroje.
<p><i>Popis</i></p>	<p>Počas tejto 95-minútovej výučby sa žiaci stredných škôl oboznámia s pokročilými koncepciami súvisiacimi s dedičstvom pomocou nástroja Greenfoot. Hodina sa začína 15-minútovým úsekom, v ktorom sa skúmajú bežné vlastnosti súvisiace s pohybom entít so zameraním na triedy Bullet a Enemy. Tieto triedy sa počas životného cyklu správajú podobne, pohybujú sa rovnakým spôsobom a potom reagujú na okolie.</p> <p>Na základe zistených spoločných vlastností bola v ďalšej 15-minútovej úlohe implementovaná nová abstraktná trieda MovingActor obsahujúca metódu <code>act()</code>. Táto trieda je spoločným predkom, ktorý bude implementovať metódu <code>act()</code>, aby sa pohyboval rovnakým spôsobom a aby sa podtriedy zamerali na svoj špecifický účel. Okrem toho sa trieda MovingActor vytvorila ako predok tried Bullet a Enemy, čím sa triedy Bullet a Enemy stali jej potomkami.</p> <p>V ďalšom 15-minútovej časti sa skúmajú vlastnosti špecifické pre triedy súvisiace s pohybom entít. V tejto súvislosti sa skúma metóda <code>act()</code> príslušných tried, ako aj atribúty <code>moveDelay</code> a <code>nextMoveCounter</code>. Možno konštatovať, že kód metódy <code>act()</code> zodpovednej za pohyb je rovnaký.</p> <p>Potom nasleduje 20-minútový blok, počas ktorého bolo predstavené kľúčové slovo <code>super</code> v kontexte dedičnosti.</p> <p>V poslednom 30-minútovom úseku sa vykonala refaktorizácia kódu súvisiaca s pohybom entít. Výsledkom je, že predtým identifikované atribúty <code>moveDelay</code> a <code>nextMoveCounter</code> z podtried Bullet a Enemy sa presunuli do predka triedy MovingActor. Okrem toho je v triede MovingActor definovaný parametrický konštruktor na inicializáciu týchto atribútov. Tento konštruktor s príslušnými parametrami bol vyvolaný z podtried Bullet a Enemy pomocou kľúčového slova <code>super</code>. Okrem toho bol kód zodpovedný za pohyb v metóde <code>act()</code> podtried Bullet a Enemy presunutý do metódy <code>act()</code> triedy MovingActor, pričom zvyšok implementácie zostal v podtriedach nezmenený. Nakoniec sa rodičovská verzia metódy <code>act()</code> volá ako prvý riadok</p>

	<p>metódy <code>act()</code> v podtriedach Bullet a Enemy.</p> <p>Výsledkom je, že na konci hodiny sa žiaci oboznámia s novými pojmami súvisiacimi s dedičnosťou.</p>
<i>Hodnotenie</i>	<p>Gamifikácia predstavuje neformálne hodnotenie, ale zvýši záujem, vnútornú motiváciu a výsledky vzdelávania celej skupiny.</p> <p>Vzhľadom na dôležitosť konceptov dedičnosti otvára štruktúra projektu možnosti pre ďalšiu diskusiu a modifikáciu. V tejto súvislosti možno uvažovať o viacerých triedach a ich hierarchii, zaviesť ďalšie triedy, metódy a atribúty. Na druhej strane môže učiteľ túto tému upraviť tak, aby ukázal výhody dedičnosti a s ňou spojenú univerzálnosť len na tu navrhnutých hierarchiách.</p>
<i>Šírenie výsledkov</i>	<p>Na šírenie výsledkov medzi učiteľmi a spolužiakmi sa použije bežné nastavenie Learning management systému (napr. Moodle). Žiaci môžu pokračovať v diskusii na danú tému na fóre, ktoré im je poskytnuté prostredníctvom nástroja na Learning management.</p>

8.2.1. Príručka učiteľa na prípravu vyučovacej hodiny

1. Úloha 6.4 - Identifikácia spoločných vlastností tried **Bullet** a **Enemy**

Cieľ:

Žiaci skúmajú triedy **Bullet** a **Enemy** s dôrazom na ich podobné správanie počas života, najmä na to, ako sa pohybujú a reagujú na okolie.

Pojmy na diskusiu:

Dedičnosť, hierarchia tried, abstraktné triedy.

Aktivita:

Zameriavame sa na triedy **Bullet** a **Enemy**, ktoré sa počas života správajú podobne. Treba si všimnúť, že tieto triedy sa pohybujú rovnakým spôsobom a potom reagujú na okolie.

2. Úloha 6.5 - Pridanie abstraktnej triedy **MovingActor** ako predka tried **Bullet** a **Enemy**

Cieľ:

Skúmanie tried **Bullet** a **Enemy** s dôrazom na ich podobné správanie počas života, najmä na to, ako sa pohybujú a reagujú na okolie.

Pojmy na diskusiu:

Dedičnosť, hierarchia tried, abstraktné triedy, implementácia abstraktnej triedy pri vývoji hry.

Aktivita:

Na základe identifikovaných spoločných vlastností by mala byť implementovaná nová abstraktná trieda **MovingActor** obsahujúca metódu **act()**. Okrem toho sa trieda **MovingActor** vytvorí ako predok tried **Bullet** a **Enemy**, čím sa triedy **Bullet** a **Enemy** stanú jej potomkami. Malo by sa diskutovať o tom, že podtriedy dedia spoločné vlastnosti z rodičovskej triedy. trieda **MovingActor** je vzorom pre návrh triedy a mala by sa deklarovať ako abstraktná.

Commit: [43e53b533563ce0a860b294ad9009f77409c48d4](https://github.com/43e53b533563ce0a860b294ad9009f77409c48d4)

3. Úloha 6.6. - Identifikácia atribútov tried **Bullet** a **Enemy** potrebných na pohyb

Cieľ:

Skúmanie tried **Bullet** a **Enemy** s dôrazom na ich podobné správanie počas života, najmä na to, ako sa pohybujú a reagujú na okolie.

Pojmy na diskusiu:

Dedičnosť, hierarchia tried, abstraktné triedy.

Aktivita:

Skúmajú sa vlastnosti špecifické pre triedu, ktoré súvisia s pohybom entít. Skúma sa metóda **act()** príslušných tried, ako aj atribúty **moveDelay** a **nextMoveCounter**. Možno konštatovať, že zdrojový kód metódy **act()** zodpovednej za pohyb je rovnaký.

4. Úvod do kľúčového slova **super** v kontexte dedičnosti

Cieľ:

Predstavenie kľúčového slova **super** v kontexte dedičnosti, demonštrácia jeho použitia na vyvolanie vlastností z rodičovskej triedy a diskusia o jeho výhodách.

Pojmy na diskusiu:

Dedičnosť, hierarchia tried, kľúčové slovo **super** v kontexte dedičnosti. Pozícia príkazu **super**.

Aktivita:

Učiteľ predstaví kľúčové slovo **super**. Kľúčové slovo **super** v kontexte dedičnosti bolo predstavené:

- kľúčové slovo **super** možno použiť na vyvolanie konštruktora z nadradenej triedy,
- kľúčové slovo **super** možno použiť na vyvolanie metódy z nadradenej triedy,
- kľúčové slovo **super** možno použiť na vyvolanie atribútu z nadradenej triedy,
- kľúčové slovo **super** musí byť použité ako prvý príkaz v konštruktore.

Je potrebné diskutovať o výhodách používania kľúčového slova **super** v kontexte dedičnosti.

5. Úloha6.7. - Presun kódu zodpovedného za pohyb do triedy MovingActor

Ciel:

Refaktorovanie kódu súvisiaceho s pohybom entít z podtried **Bullet** a **Enemy** do predchodcu triedy **MovingActor**.

Pojmy na diskusiu:

Dedičnosť, hierarchia tried, kľúčové slovo `super` v kontexte dedičnosti.

Aktivita:

Bola vykonaná refaktorizácia kódu týkajúca sa pohybu entít. Doteraz identifikované atribúty `moveDelay` a `nextMoveCounter` z podtried **Bullet** a **Enemy** sa presunuli do predka triedy **MovingActor**. Parametrický konštruktor na inicializáciu týchto atribútov je definovaný v triede **MovingActor**. Tento konštruktor s príslušnými parametrami bol vyvolaný z podtried **Bullet** a **Enemy** pomocou kľúčového slova `super`. Kód zodpovedný za pohyb v metóde `act()` podtried **Bullet** a **Enemy** bol presunutý do metódy `act()` triedy **MovingActor**, zatiaľ čo zvyšok implementácie zostal v podtriedach nezmenený. Nakoniec sa rodičovská verzia metódy `act()` volá ako prvý riadok metódy `act()` v podtriedach **Bullet** a **Enemy**. treba diskutovať o tom, že podtriedy môžu obsahovať ďalšie vlastnosti, ktoré nie sú dostupné v rodičovskej triede (t. j. iná implementácia metódy `act()`).

Commit: [ca1f010a63445c1847b74259a1c6cd4817121db3](#)

8.3. Koncepty dedičnosti v prostredí Greenfoot (časť 2)

Tabuľka 20: Koncepty dedičnosti v prostredí Greenfoot

Názov	Koncepty dedičnosti v prostredí Greenfoot
Výchovno-vzdelávacie ciele a ich opis	Na konci tejto časti budú žiaci rozumieť ďalším konceptom dedičnosti . Preskúmané pojmy sa budú rozoberať v kontexte vývoja hier, čím sa podporí tvorivé myslenie, tímová práca a nadšený prístup ku kódovaniu v prostredí Greenfoot.
Cieľová skupina	Žiaci stredných škôl, ktorí sa zúčastňujú kurzu OOP4Fun. Základné znalosti programovania a základné znalosti objektovo orientovaného programovania. Žiaci by mali byť oboznámení s programom Greenfoot.
Časové trvanie	1. Úloha 6.8 - Vytvorenie vlastných nepriateľov (30 minút) 2. Úvod do Liskovej princípu substitúcie (20 minút)

	3. Úloha 6.9 - Vytváranie vlastných nepriateľov (20 minút)
<i>Materiály a zdroje</i>	<ul style="list-style-type: none"> • Učebnica z projektu OOP4Fun. • Zdroje z projektu OOP4Fun. • Zdrojový kód projektu z Github/Gitlab. • Internetové zdroje.
<i>Popis</i>	<p>Počas tejto 70-minútovej výučby sa žiaci stredných škôl oboznámia s pokročilými koncepciami dedičnosti pomocou nástroja Greenfoot. Lekcia sa začína 30-minútovým úsekom zameraným na triedu Enemy (Nepriateľ), kde žiaci vytvárajú ďalšie podtriedy predstavujúce rôznych nepriateľov (napr. Frog (Žaba) a Spider (Pavúk)). V tejto súvislosti sú pre každý typ nepriateľa definované obrázky a bezparametrické konštruktory (s príslušným volaním rodičovského konštruktora). V ďalšom 20-minútovom bloku sa predstaví Liskov princíp substitúcie (LSP).</p> <p>Tento princíp, ktorý je súčasťou princípov SOLID objektovo orientovaného návrhu, uvádza, že funkcie, ktoré používajú ukazovatele alebo odkazy na nadradené triedy, by mali byť schopné používať objekty podtried.</p> <p>Následne je 20-minútová úloha venovaná vytváraniu vlastných nepriateľov. Skúma sa metóda Arena.spawn() a prostredníctvom rôznych rozhodnutí sa vytvárajú vlastní nepriatelia a ukladajú sa do premennej typu Enemy. Pozoruje sa, že v aplikácii nie je potrebné meniť žiadny iný kód, čo dokazuje aplikáciu LSP.</p> <p>Výsledkom je, že na konci hodiny sa žiaci oboznámia s pokročilými konceptmi dedičnosti a praktickými aplikáciami.</p>
<i>Hodnotenie</i>	<p>Gamifikácia predstavuje neformálne hodnotenie, ale zvýši záujem, vnútornú motiváciu a výsledky vzdelávania celej skupiny.</p> <p>Vzhľadom na dôležitosť konceptov dedičnosti otvára štruktúra projektu možnosti pre ďalšiu diskusiu a modifikáciu. V tejto súvislosti možno uvažovať o ďalších triedach a ich hierarchii a zaviesť ďalšie triedy, metódy a atribúty. Na druhej strane môže učiteľ túto tému upraviť tak, aby ukázal výhody dedičnosti a s ňou spojenú univerzálnosť len na tu navrhnutých hierarchiách.</p>
<i>Šírenie</i>	Na šírenie výsledkov medzi učiteľmi a spolužiakmi sa použije bežné nastavenie Learning management systému (napr. Moodle). Žiaci môžu

výsledkov

pokračovať v diskusii na danú tému na fóre, ktoré im je poskytnuté prostredníctvom nástroja na Learning management.

8.3.1. Príručka učiteľa na prípravu vyučovacej hodiny

1. Úloha 6.8 - Vytvorenie vlastných nepriateľov

Cieľ:

Žiaci sa zaoberajú skúmaním podtried triedy **Enemy**, každá s obrázkami a bezparametrickými konštruktormi, ktoré vhodne vyvolávajú rodičovský konštruktor.

Pojmy na diskusiu:

Dedičnosť, hierarchia tried, kľúčové slovo super v kontexte dedičnosti.

Aktivita:

Zameriavame sa na triedu **Enemy** a definíciu ďalších podtried reprezentujúcich rôznych nepriateľov (napr. **Frog** a **Spider**). Pre každý typ nepriateľa by mali byť definované obrázky a bezparametrické konštruktory (s príslušným volaním rodičovského konštruktora).

Commit: [b0ac1fbe793548a32f7700c292aed631918c8388](https://github.com/b0ac1fbe793548a32f7700c292aed631918c8388)

2. Úvod do Liskovej princípu substitúcie

Cieľ:

Predstavenie Liskovej princípu substitúcie, diskusia o reálnom svete a skúmanie výhod dodržiavania tohto princípu v kontexte dedenia.

Pojmy na diskusiu:

Dedičnosť, hierarchia tried, referenčné premenné, Liskovej princípu substitúcie.

Aktivita:

Zavádza sa princíp Liskovej substitúcie. Tento princíp je súčasťou SOLID princípov objektovo orientovaného návrhu. Princíp uvádza, že funkcie, ktoré používajú ukazovatele alebo odkazy na nadradené triedy, by mali byť schopné používať objekty podtried. Mali by sa rozobrať reálne príklady (napr. ak je trieda **Computer** definovaná ako rodičovská trieda a triedy **Console**, **Desktop**, **Laptop** a **Mobile Phone** sú definované ako podtriedy, Liskovej princípu substitúcie hovorí, že funkcie, ktoré používajú triedu **Computer**, budú pracovať aj so všetkými podtriedami bez akejkoľvek zmeny v kóde). Mali by sa prediskutovať výhody používania Liskovej princípu substitúcie v kontexte dedičnosti.

3. Úloha 6.9 - Vytváranie vlastných nepriateľov

Cieľ:

Demonštrácia aplikácie Liskovej princípu substitúcie vytvorením vlastných nepriateľov.

Pojmy na diskusiu:

Dedičnosť, hierarchia tried, referenčné premenné, Liskovej princípu substitúcie.

Aktivita:

Táto úloha je určená na vytváranie vlastných nepriateľov. Skúma sa metóda **Arena.spawn()** a prostredníctvom rôznych rozhodnutí sa vytvárajú vlastní nepriatelia a ukladajú sa do premennej typu **Enemy**. Treba si všimnúť, že v aplikácii nie je potrebné meniť žiadny iný kód, čo dokazuje uplatnenie Liskovej princípu substitúcie.

Commit: [8cd4397f585ec957bbc18ca98e01823f434a13a6](https://github.com/0x09b4/arena/commit/8cd4397f585ec957bbc18ca98e01823f434a13a6)

8.4. Koncepty dedičnosti v prostredí Greenfoot (časť 3)

Tabuľka 21: Koncepty dedičnosti v prostredí Greenfoot

<i>Názov</i>	Koncepty dedičnosti v prostredí Greenfoot
<i>Výchovno-vzdelávacie ciele a ich opis</i>	Na konci tejto časti budú žiaci rozumieť ďalším konceptom dedičnosti . Preskúmané pojmy sa budú rozoberať v kontexte vývoja hier, čím sa podporí tvorivé myslenie, tímová práca a nadšený prístup ku kódovaniu v prostredí Greenfoot.
<i>Cieľová skupina</i>	Žiaci stredných škôl, ktorí sa zúčastňujú kurzu OOP4Fun. Základné znalosti programovania a základné znalosti objektovo orientovaného programovania. Žiaci by mali byť oboznámení s programom Greenfoot.
<i>Časové trvanie</i>	<ol style="list-style-type: none">1. Úloha 6.10 - Diskusia o hierarchii arén (20 minút)2. Úloha 6.11 - Vytvorenie univerzálnej arény (30 minút) a 6.12 - Vytvorenie triedy DemoArena (15 minút)3. Úloha 6.13 - Vytvorenie vlastných arén (30 minút)4. Opakovanie teórie dedičnosti (20 minút)
<i>Materiály a zdroje</i>	<ul style="list-style-type: none">• Učebnica z projektu OOP4Fun.• Zdroje z projektu OOP4Fun.• Zdrojový kód projektu z Github/Gitlab.• Internetové zdroje.

<p><i>Popis</i></p>	<p>Počas tejto 115-minútovej výučby sa žiaci stredných škôl oboznámia s pokročilými koncepciami dedičnosti pomocou nástroja Greenfoot. Lekcia sa začína 20-minútovou diskusiou o hierarchii arény. Podtriedy triedy Arena sú zodpovedné za vlastné rozloženia (napr. pozície inštancií Orb a Direction, veľkosť arény). Tieto úlohy sa vykonávajú v konštruktoroch podtried, ktoré nastavujú a ukladajú pozície miesta vzniku, rotácie a rozmery arény.</p> <p>V nasledujúcej 30-minútovej úlohe je zavedená univerzálna trieda Arena. Definujú sa ďalšie atribúty (spawnPositionX, spawnPositionY a spawnRotation), ktoré sa inicializujú v konštruktoze a používajú sa v metódach spawn() a respawn(Enemy). Atribúty týkajúce sa rozmerov arény (šírka a výška) sú tiež definované a inicializované v konštruktoze. Keďže trieda Arena slúži ako vzor na definovanie konkrétnych arén, je definovaná ako abstraktná trieda.</p> <p>Na základe identifikovanej triedy Arena sa v ďalšej 15-minútovej úlohe predstaví podtrieda DemoArena. Definuje sa konštruktor DemoArena, ktorý volá konštruktor nadradenej triedy, a kód zodpovedný za rozloženie smerov, inštancií triedy Orb a veží sa presunie z konšuktora Arena do konšuktora DemoArena. Nakoniec sa vytvorí nová inštancia triedy DemoArena.</p> <p>V nasledujúcom 30-minútovom úseku sa vytvoria ďalšie inovatívne podtriedy triedy Arena. Kód je možné zdieľať s ostatnými žiakmi v skupine.</p> <p>Nakoniec sa posledná 20-minútová fáza zaoberá teóriou súvisiacou s dedičnosťou.</p> <p>Výsledkom je, že na konci hodiny sa žiaci oboznámia s pokročilými konceptmi dedičnosti a praktickými aplikáciami.</p>
<p><i>Hodnotenie</i></p>	<p>Gamifikácia predstavuje neformálne hodnotenie, ale zvýši záujem, vnútornú motiváciu a výsledky vzdelávania celej skupiny.</p> <p>Vzhľadom na dôležitosť konceptov dedičnosti otvára štruktúra projektu možnosti pre ďalšiu diskusiu a modifikáciu. V tejto súvislosti možno uvažovať o ďalších triedach a ich hierarchii a zaviesť ďalšie triedy, metódy a atribúty. Na druhej strane môže učiteľ túto tému upraviť tak, aby ukázal výhody dedičnosti a s ňou spojenú univerzálnosť len na tu navrhnutých hierarchiách.</p>
<p><i>Šírenie</i></p>	<p>Na šírenie výsledkov medzi učiteľmi a spolužiakmi sa použije bežné nastavenie Learning management systému (napr. Moodle). Žiaci môžu</p>

8.4.1. Príručka učiteľa na prípravu vyučovacej hodiny

1. Úloha 6.10 - Diskusia o hierarchii arén

Cieľ:

Žiaci preskúmajú hierarchie tried **Arena** s dôrazom na to, ako sú podtriedy zodpovedné za definovanie vlastných rozložení a implementáciu týchto rozložení v rámci ich príslušných konštruktorov.

Pojmy na diskusiu:

Dedičnosť, hierarchia tried, konštruktory.

Aktivita:

Hierarchia tried arény je prediskutovaná, Je potrebné poznamenať, že podtriedy arény sú zodpovedné za vlastné rozloženie (napr. pozície inštancií **Orb** a **Direction**, veľkosť arény). Tieto úlohy sa vykonávajú v konštruktoroch podtried, ktoré nastavujú a ukladajú pozície miest vzniku nepriateľov, rotácie a rozmery arény.

2. Úloha 6.11 - Vytvorenie univerzálnej arény (30 minút) a 6.12 - Vytvorenie triedy DemoArena

Cieľ:

Predstavenie univerzálnej abstraktnej triedy **Arena**, definovanie a inicializácia konkrétnej podtriedy **Arena**, preskúmanie hierarchie tried **Arena**.

Pojmy na diskusiu:

Dedičnosť, hierarchia tried, abstraktné triedy, konštruktory.

Aktivita:

Na základe predchádzajúcej diskusie sa zavádza univerzálna trieda **Arena**. Sú definované ďalšie atribúty (**spawnPositionX**, **spawnPositionY** a **spawnRotation**), ktoré sú inicializované v konštruktoze a použité v metódach **spawn()** a **respawn(Enemy)**. Atribúty týkajúce sa rozmerov arény (šírka a výška) sú tiež definované a inicializované v konštruktoze. Keďže trieda **Arena** slúži ako vzor na definovanie konkrétnych arén, je definovaná ako abstraktná trieda.

Commit: [e9844d7d9b5f19969618b469ebc907d0fe3c1357](https://github.com/epicurean/e9844d7d9b5f19969618b469ebc907d0fe3c1357)

Na základe identifikovanej triedy **Arena** je definovaná podtrieda **DemoArena**. Definuje sa konštruktor **DemoArena**, ktorý volá konštruktor nadradenej triedy, a kód zodpovedný za rozloženie smerov, inštancaie triedy **Orb** a veží sa presunie z konštruktoza triedy **Arena** do

konštruktora **DemoArena**. Nakoniec sa vytvorí nová inštancia triedy **DemoArena**. Ak chcete vytvoriť inštanciu triedy **DemoArena**, kliknite na túto triedu pravým tlačidlom myši a vyberte položku **new DemoArena**.

Commit: [6a6569774b5735f453a56c7cb2cdbf19d228eae9](#)

3. Úloha 6.13 - Vytvorenie vlastných arén

Cieľ:

Definovanie a inicializácia vlastných podtried **Arena**, skúmanie hierarchie tried **Arena**.

Pojmy na diskusiu:

Dedičnosť, hierarchia tried, abstraktné triedy, konštruktory.

Aktivita:

Vytvorí sa ďalšie inovatívne podtriedy triedy **Arena**. Kód je možné zdieľať s ostatnými žiakmi v skupine.

4. Opakovanie teórie dedičnosti

Cieľ:

Prehľad koncepcie a výhod dedičnosti, diskusia o hierarchii tried a abstraktných triedach, skúmanie kľúčového slova **super** a Liskovej princípu substitúcie spolu s ich výhodami a skúmanie príkladov a implementácií dedičnosti z reálneho života a hier.

Pojmy na diskusiu:

Dedičnosť, hierarchia tried, abstraktné triedy, kľúčové slovo **super**, Liskovej princípu substitúcie, príklady a implementácie dedičnosti z reálneho života a hier.

Aktivita:

Preskúma sa pojem dedičnosti. Preskúmajú sa výhody dedenia. Rozoberá sa hierarchia tried a jej výhody v kontexte koncepcie dedičnosti. Preberá sa pojem abstraktnej triedy. Rozoberá sa kľúčové slovo **super** a jeho výhody v kontexte dedičnosti. Je preskúmaný Liskovej princípu substitúcie a výhody používania Liskovej princípu substitúcie v kontexte dedičnosti. Diskutuje sa o príkladoch dedičnosti z reálneho života. diskutuje sa o príkladoch dedičnosti v hrách a ich implementácií.

9. Zapuzdrenie

V rámci tematického celku Zapuzdrenie boli vytvorené dva metodické materiály.

9.1. Skúmanie zapuzdrenia prostredníctvom vývoja hier s Greenfoot

Tabuľka 22: Skúmanie zapuzdrenia prostredníctvom vývoja hier s Greenfoot

<i>Názov</i>	Skúmanie zapuzdrenia prostredníctvom vývoja hier s Greenfoot
<i>Výchovno-vzdelávacie ciele a ich opis</i>	Žiaci porozumejú zapuzdreniu prostredníctvom ďalšieho vývoja hry TowerDefense.
<i>Cieľová skupina</i>	Žiaci stredných škôl, ktorí sa zúčastňujú kurzu OOP4Fun. Základné znalosti programovania vrátane dedičnosti. Žiaci by sa mali oboznámiť so systémom Greenfoot vo všeobecnosti.
<i>Časové trvanie</i>	<ol style="list-style-type: none">1. Úvod (5 minút)2. Úloha 7.1 - Vytvorenie triedy ManualTower (20 minút)3. Úloha 7.2 - Zmena ovládania manuálne ovládanej veže a 7.3 - Vyvolanie zmeny manuálneho ovládania (30 minút)4. Diskusia (35 minút)5. Vysvetlenie kódu (25 minút)6. Úloha 7.4 - Spracovanie ovládania používateľa (10 minút)
<i>Materiály a zdroje</i>	<ul style="list-style-type: none">• Učebnica z projektu OOP4Fun.• Zdroje z projektu OOP4Fun.• Zdrojový kód projektu z Github/Gitlab.• Internetové zdroje.
<i>Popis</i>	<p>Na tomto stretnutí sa žiaci stredných škôl dozvedia o zapuzdrení v rámci objektovo orientovaného programovania (OOP) v Greenfoot. Lekcia sa začína stručným 5-minútovým úvodom, v ktorom učiteľ načrtne ciele.</p> <p>Žiaci začnú vývojom triedy <code>ManualTower</code> ako podtriedy triedy <code>Tower</code>. Táto aktivita sa zameriava na definovanie dvoch konštruktorov, ktoré sú v súlade s konštruktormi rodičovskej triedy, aby sa zabezpečila správna inicializácia. Mali by implementovať metódu <code>act()</code>, ktorá najprv zavolá</p>

	<p>metódu act() rodičovskej triedy.</p> <p>Po vytvorení tejto triedy učiteľ zavedie atribút typu boolean isManuallyControlled, ktorý inicializuje na hodnotu false. Žiaci vytvoria metódu changeControl(boolean), ktorá prepína stav isManuallyControlled a podľa toho mení obraz veže, čím demonštrujú zapuzdrenie riadením prístupu k stavu objektu prostredníctvom metód. Každý žiak by potom mal ručne spustiť metódu changeControl() na inštanciách objektu ManualTower a pozorovať, ako sa zmení vnútorný stav a vonkajšie zobrazenie.</p> <p>Jadrom lekcie je vytvorenie súkromnej metódy processUserControl(), ktorá by mala detegovať kliknutia myšou na inštanciu veže. Po kliknutí metóda zmení stav ovládania veže a aktualizuje jej orientáciu na základe polohy myši, pričom použije zapuzdrenie na skrytie zložitej logiky ovládania. Žiaci by mali implementovať metódu a integrovať ju do metódy act(), testovať interakciu s herným prostredím, aby sa zabezpečila funkčnosť, a naučiť sa, ako súkromné metódy chránia kód pred vonkajšími zmenami.</p>
<p><i>Hodnotenie</i></p>	<p>Táto aktivita umožní učiteľom poskytovať formatívnu spätnú väzbu na základe diskusií a monitorovania obrátenej triedy a tímovej práce žiakov.</p> <p>Vzájomné hodnotenie sa bude vykonávať online ako súčasť domácej úlohy. Žiakom sa tak pripomenú dôležité aspekty cvičenia, donútia sa kriticky posúdiť prácu ostatných žiakov, získajú prehľad o dobrých či menej dobrých riešeniach svojich kolegov atď. a zvýši sa celkové dosiahnutie výsledkov vzdelávania. Pri práci v tímovom projekte, na ktorom žiaci pracujú, sa budú využívať aj tieto výsledky vzdelávania a poznatky.</p>
<p><i>Šírenie výsledkov</i></p>	<p>Na šírenie výsledkov medzi učiteľmi a spolužiakmi sa použije bežné nastavenie Learning management systému (napr. Moodle). Žiaci môžu pokračovať v diskusii na danú tému na fóre, ktoré im je poskytnuté prostredníctvom nástroja na Learning management.</p>

9.1.1. Príručka učiteľa na prípravu vyučovacej hodiny

1. Úvod

Učiteľ by mal spustiť predtým vytvorenú hru a pozorovať, ako sa správajú rôzni aktéri. Navrhnete vytvorenie ďalšieho typu veže, ktorú možno ovládať ručne, aby sa ľahšie odstraňovali nepriatelia. Používateľ by mal mať možnosť ovládať vždy jednu vežu. Keď sa

na vežu klikne, mala by sa stať ručne ovládanou. Na označenie toho, ktorá veža je manuálne ovládaná, by mala mať aktuálne ovládaná veža iný vzhľad.

2. Úloha 7.1 - Vytvorenie triedy `ManualTower`

Cieľ:

Žiaci si pripravíajú triedu `ManualTower`.

Pojmy na diskusiu:

Dedičnosť, triedy, konštruktory.

Aktivita:

Keďže žiaci už vedia, ako vytvoriť triedu potomka, nechajte ich vytvoriť tímy a vytvoriť triedu `ManualTower` ako potomka triedy `Tower`. Žiaci by mali implementovať konštruktory aj metódu `act()` a zabezpečiť, aby sa z týchto metód volali konštruktory nadradenej triedy. V tejto časti hodiny si žiaci zopakujú prebranú látku, použijú ju a zdokonalia svoje praktické znalosti o dedičnosti.

Commit: [63a02fa0c5080165cba8b467da08c4b65f31d0a8](https://commit-hash.com/63a02fa0c5080165cba8b467da08c4b65f31d0a8)

3. Úloha 7.2 - Zmena ovládania manuálne ovládanej veže a 7.3 - Vyvolanie zmeny manuálneho ovládania

Cieľ:

Učiteľ vysvetlí žiakom potrebu súkromných metód definovaných funkcie `changeControl()`.

Pojmy na diskusiu:

Metódy, triedy, atribúty, modifikátory prístupu.

Aktivita:

Učiteľ by mal pripraviť ikony pre ručne ovládanú vežu. Na programovú zmenu ikony objektu by mal učiteľ žiakom vysvetliť, ako sa používa metóda `Actor.setImage(String)`. Nechajte žiakom určitý čas na vyskúšanie tejto funkcie.

Učiteľ by mal so žiakmi prediskutovať, ako zistiť, či je veža ovládaná ručne. Zdôraznite, že nie je dôležité len zmeniť stav objektu, ale aj aktualizovať jeho obraz. Zdôraznite, že ak chce používateľ zmeniť stav objektu veže a zmení priamo len atribút, obrázok zostane rovnaký. Táto diskusia by mala žiakom pomôcť pochopiť potrebu meniť hodnotu atribútu prostredníctvom metódy a udržiavať atribúty skôr súkromné ako verejné. Vysvetlite žiakom, že tento postup sa nazýva zapuzdrenie, pri ktorom je vnútorný stav skrytý a verejné metódy sa používajú na kontrolovanú zmenu tohto stavu.

Umožnite žiakom implementovať logiku funkcie. Nechajte ich ručne zavolať svoju metódu a pozorovať zmeny vnútorného stavu.

Commit: [2257746b7dac5eaab7acc55d6493319230338f3a](https://commit-hash.com/2257746b7dac5eaab7acc55d6493319230338f3a)

4. Diskusia

Cieľ:

Porozumenie zapuzdreniu logiky vo vnútri samostatnej metódy.

Pojmy na diskusiu:

Metódy, vetvenie.

Aktivita:

Učiteľ by mal poukázať na to, že stav veže sa dá zmeniť len ručným vyvolaním metódy. Poukázať na to, že myš sa môže nachádzať mimo sveta, v takom prípade bude informácia o myši nulová. Pripomeňte žiakom, že metóda **act()** beží počas hry neustále a že by mala kontrolovať, či bolo na objekt kliknuté, a až potom zavolať metódu **changeControl()**. Zdôraznite, že logika spracovania ovládacieho prvku by mala byť zapuzdrená vnútri samostatnej metódy **processUserControl()**.

5. Vysvetlenie kódu

Cieľ:

Zavedenie metódy potrebnej na vyriešenie problému.

Pojmy na diskusiu:

Metódy, prostredie Greenfoot.

Aktivita:

Zvážte, ako zmeniť stav aktéra kliknutím na objekt. Na implementáciu tohto postupu je potrebné vysvetliť metódu **GreenFoot.mouseClicked(Object)**. Taktiež predstavte objekt **MouseInfo**, ktorý možno použiť na získavanie informácií o polohe myši.

6. Úloha 7.4 - Spracovanie ovládania používateľa

Cieľ:

Zlepšenie pochopenia súkromných metód a zapuzdrenia prostredníctvom praktického zadania.

Pojmy na diskusiu:

Metódy, modifikátory prístupu, triedy.

Aktivita:

Po definovaní súkromnej metódy **processUserControl()** nechajte žiakov implementovať jej logiku. Po kliknutí myšou by sa mala zmeniť ovládaná veža. Ak je veža ovládaná ručne, mala by nasledovať a smerovať k myši. Pripomeňte im, že je možné, aby myš bola mimo sveta. Po rozdelení žiakov do skupín ich nechajte implementovať logiku metódy **processUserControl()**.

Jeden alebo dva tímy odprezentujú svoju prácu a skupina bude diskutovať o výsledkoch spolu s učiteľom. Na konci hodiny by mali všetci žiaci pochopiť, ako sa táto metóda implementuje.

Commit: [6ec1f489576019a6493490f9e97797920b923869](https://github.com/6ec1f489576019a6493490f9e97797920b923869)

9.2. Skúmanie zapuzdrenia prostredníctvom vývoja hier s Greenfoot

Tabuľka 23: Skúmanie zapuzdrenia prostredníctvom vývoja hier s Greenfoot

<i>Názov</i>	Skúmanie zapuzdrenia prostredníctvom vývoja hier s Greenfoot
<i>Výchovno-vzdelávacie ciele a ich opis</i>	Žiaci porozumejú zapuzdreniu prostredníctvom ďalšieho vývoja hry TowerDefense.
<i>Cieľová skupina</i>	Žiaci stredných škôl, ktorí sa zúčastňujú kurzu OOP4Fun. Základné znalosti programovania vrátane dedičnosti. Žiaci by sa mali oboznámiť so systémom Greenfoot vo všeobecnosti.
<i>Časové trvanie</i>	<ol style="list-style-type: none"> 1. Prevrátená trieda (30 minút) 2. Atribúty triedy (5 minút) 3. Úloha 7.6 - Pridanie evidencie ručne riadenej veže (5 minút) 4. Metóda triedy (10 minút) 5. Úloha 7.7 - Zmena manuálne ovládanej veže z centralizovaného miesta (20 minút) 6. Úloha 7.8 - Vyvolanie zmeny ručne ovládanej veže (15 min) 7. Opakovanie teórie (10 min)
<i>Materiály a zdroje</i>	<ul style="list-style-type: none"> • Učebnica z projektu OOP4Fun. • Zdroje z projektu OOP4Fun. • Zdrojový kód projektu z Github/Gitlab. • Internetové zdroje.
<i>Popis</i>	Na začiatku hodiny žiaci diskutujú o problémoch súvisiacich s ovládaním používateľa, ako je napríklad nemožnosť zrušiť výber veže po jej výbere, a hľadajú ich. Žiaci by sa mali zapojiť do diskusie a navrhnúť riešenia na sledovanie aktuálne ovládanej veže a upraviť triedu <code>ManualTower</code> tak, aby obsahovala mechanizmus na zrušenie výberu veže.

	<p>Nakoniec by sa mala implementovať metóda triedy changeControlledInstance(), ktorá umožňuje zmenu ovládania veží z centralizovanej metódy, čím sa zlepší pochopenie zapuzdrenia tým, že sa ukáže, ako môžu metódy triedy spravovať spoločný stav medzi inštanciami.</p> <p>Tento komplexný vzdelávací prístup učí koncept zapuzdrenia a demonštruje jeho dôležitosť a užitočnosť v reálnych aplikáciách, čím rozvíja u žiakov schopnosti riešiť problémy a spolupracovať.</p>
<i>Hodnotenie</i>	<p>Táto aktivita umožní učiteľom poskytovať formatívnu spätnú väzbu na základe diskusií a monitorovania obrátenej triedy a tímovej práce žiakov.</p> <p>Práca v tímovom projekte, na ktorom žiaci pracujú, bude tiež využívať tieto výsledky vzdelávania a poznatky.</p>
<i>Šírenie výsledkov</i>	<p>Na šírenie výsledkov medzi učiteľmi a spolužiakmi sa použije bežné nastavenie Learning management systému (napr. Moodle). Žiaci môžu pokračovať v diskusií na danú tému na fóre, ktoré im je poskytnuté prostredníctvom nástroja na Learning management.</p>

9.2.1. Príručka učiteľa na prípravu vyučovacej hodiny

1. Prevrátená trieda

Cieľ:

Žiaci by mali rozpoznať potrebu inicializácie atribútu na jednom mieste.

Pojmy na diskusiu:

Atribúty triedy.

Aktivita:

Na začiatku hodiny nechajte žiakov identifikovať problém s ovládaním používateľa (úloha 7.5). V súčasnosti nie je možné zrušiť výber veže. Vyzvite žiakov, aby sa zamysleli nad tým, ako by sa tento problém dal vyriešiť. Vysvetlite, že v hre by mala byť vždy vybraná len jedna veža. táto diskusia by mala žiakov priviesť k myšlienke mať v programe jedno miesto, ktoré sa inicializuje len raz a je k nemu prístup z iných častí programov, z iných objektov a aktérov.

2. Atribúty triedy

Cieľ:

Predstavenie základov atribútov tried

Pojmy na diskusiu:

Atribúty, atribúty tried, triedy.

Aktivita:

Vysvetlite, čo sú atribúty triedy: premenné, ktoré patria samotnej triede, a nie inštanciam triedy. Dajte tento koncept do súvislosti so scenárom hry, o ktorom sme hovorili skôr, kde by sa problém mohol vyriešiť tým, že by sa centrálnne spravoval atribút aktuálne vybranej veže.

3. Úloha 7.6 - Pridanie evidencie ručne riadenej veže

Cieľ:

Praktické použitie atribútov tried a **null**.

Pojmy na diskusiu:

Atribúty, atribúty tried, triedy, modifikátory prístupu.

Aktivita:

Ak chcete sledovať, ktorá veža je v hre aktuálne vybraná, pridajte do triedy **ManualTower** súkromný statický atribút **controlledInstance** a inicializujte ho na **null**. Statický atribút sa vzťahuje na celú triedu, nie na objekt triedy. Preto nám definovanie statickej premennej umožní zistiť, či bola vybraná veža, a ak áno, ktorá, odkazom na názov triedy bez toho, aby sme museli pristupovať ku konkrétnemu objektu. Učiteľ by mal zdôrazniť, že pre celú hru existuje jedna riadená inštancia. Na začiatku by mala byť riadená inštancia inicializovaná na **null**, pretože nie je vybraná žiadna veža. Skontrolujte vnútorný stav triedy. Tu učiteľ vysvetlí rozdiely medzi statickými a nestatickými atribútmi. Učiteľ so žiakmi diskutuje o výhodách používania statických atribútov v hrách. Učiteľ by tu mal spomenúť aj statické metódy a diskutovať so žiakmi o tom, v čom je používanie statických metód výhodné.

Commit: [c4739460bed583d2126de066acc6b1149d022990](https://github.com/4739460bed583d2126de066acc6b1149d022990)

4. Metóda triedy

Cieľ:

Predstavenie základov metód triedy.

Pojmy na diskusiu:

Metódy triedy, trieda, objekty.

Aktivita:

Predstavte koncept metód triedy, ktoré môžu pracovať s údajmi na úrovni triedy. Diskutujte o potrebe metód ako **changeControlledInstance** na riadenie prepínania aktuálne riadenej veže. Zdôraznite, že tieto metódy možno volať bez toho, aby ste potrebovali inštanciu triedy. Napríklad školský zvonček zvoní pre všetkých v rovnakom

čase, nezáleží na tom, kto ste, na druhej strane kontrola domácej úlohy žiaka vyžaduje informácie o konkrétnom žiakovi.

5. Úloha 7.7 - Zmena manuálne ovládanej veže z centralizovaného miesta

Cieľ:

Praktické použitie metód triedy.

Pojmy na diskusiu:

Metódy, metódy triedy, atribúty triedy.

Aktivita:

Učiteľ by mal pridať metódu **changeControlledInstance** na zmenu ručne ovládanej veže. Parametrom metódy bude veža, ktorú chce používateľ vybrať. Najskôr by sa malo skontrolovať, či je ovládaná inštancia aktuálne vybraná. Ak je, nemalo by sa nič meniť, ale ak je odovzdaná inštancia iná, mali by sme zmeniť aktuálne ovládanú inštanciu (mal by sa zmeniť odkaz na aktuálne ovládanú inštanciu). Metódu vyskúšajte ručne a pozorujte, že ikony veží sa nemenia. Poukážte na to, že iba zmena referencie na riadenú inštanciu by nezmenila ovládanie a že by sa to malo urobiť ručne. Pridajte kód, ktorý uvoľní aktuálne ovládanú inštanciu, a po aktualizácii referencie pridajte kód, ktorý nastaví ručné ovládanie novo ovládanej inštancie. Zdôraznite potrebu kontroly **null** referencií, ktoré by sa mohli objaviť, ak neexistuje aktuálne kontrolovaná inštancia a ak neexistuje novo kontrolovaná inštancia (keď je parameter **null**).

Commit: [9dc6d8dd4dcbbd71edb8009c1a72403dea1a0ee0](https://github.com/9dc6d8dd4dcbbd71edb8009c1a72403dea1a0ee0)

6. Úloha 7.8 - Vyvolanie zmeny ručne ovládanej

Cieľ:

Praktické použitie metód triedy.

Pojmy na diskusiu:

Metódy, metódy triedy, atribúty triedy.

Aktivita:

Ručne otestujte, či funkcia funguje správne. Potom so žiakmi prediskutujte, kde by sa mala táto funkcia zavolať. Metóda by mala byť vyvolaná vo vnútri metódy **act()** v triede **Arena** a vo vnútri funkcie **processUserController()**. Nakoniec urobte metódu **ManualTower.changeControl(boolean)** súkromnou a pozorujte zmeny inštancie **ManualTower**.

Commit: [c052bbb6aa4c7e690d4d8cf55d3831028fa2b9e3](https://github.com/c052bbb6aa4c7e690d4d8cf55d3831028fa2b9e3)

7. Opakovanie teórie

Zhrňte túto lekciiu a zdôraznite dôležitosť atribútov a metód tried pri efektívnom riadení hernej logiky. Povzbudte žiakov k ďalšiemu skúmaniu aplikovaním týchto konceptov vo vlastných programátorských projektoch.