

Domácí úloha 08 (17.11. až 28.11.2021 – 4 body) – ppa1u08.jar

- Napište jednoduchou interaktivní hru, která umožní postavě projít bludiště.
- Aplikace se bude skládat ze tříd a veřejných metod uvedených níže.
 - o Případné další soukromé třídy či metody vytvářejte dle potřeby.
- Pomocí dokumentačních komentářů úlohu řádně okomentujte.
 - o Komentář by měl být u každé třídy, metody a atributu (třídy i instance).
 - o Pro třídy uveďte i autora a verzi (jako verzi uveďte aktuální datum ve tvaru RRMMDD).
- Výčet `Smer`
 - o Výčet slouží k uložení informace o světové straně:
 - `SEVER` – nahoru
 - `VYCHOD` – vpravo
 - `JIH` – dolů
 - `ZAPAD` – vlevo
- Třída `GUI`
 - o Třída se stará o správné vykreslení světa a hráče a o načítání ovládání.
 - o Protože se musí kreslit více informací přes sebe, zapisují se data nejprve na plátno (zvolte si vhodnou vnitřní reprezentaci, např. pole znaků) a následně se vykreslí celé plátno na `System.out`.
 - o Třída v konstruktoru přebírá referenci na `Scanner`, aby mohla načítat informace z klávesnice.
 - o Veřejné metody
 - `GUI(int sirka, int vyska, Scanner sc)`
 - Konstruktor připraví plátno o rozměrech šířka x výška a uchová si referenci na `Scanner`.
 - `void smaz()`
 - Metoda vyčistí plátno, tj. všechny prvky plátna budou mít hodnotu znak ' ' (mezera).
 - `void zapis(int x, int y, char znak)`
 - Metoda zapíše na plátno na pozici [x; y] daný znak.
 - `void vykresli()`
 - Metoda vypíše jednotlivé znaky plátna na `System.out`.
 - `Smer nactiAkci()`
 - Metoda načte pomocí `Scanneru` (parametr konstruktoru) celé číslo a podle jeho hodnoty vrátí odpovídající směr.
 - Zeměpisný směr odpovídá číslům na numerické klávesnici, tedy:
 - o 8 – SEVER
 - o 6 – VYCHOD
 - o 2 – JIH
 - o 4 – ZAPAD
 - Při zadání jiného čísla metoda vrátí `null`.
- Třída `Svet`
 - o Třída reprezentuje svět (bludiště).
 - o Informace o světě jsou předány po řádkách v poli znaků.
 - Znak ' ' (mezera) určuje místa, kam lze dojít.

- Znak '#' určuje místo, kam dojít nelze.
- Veřejné metody
 - `Svet(int sirka, int vyska, char[] data)`
 - Konstruktor vytvoří reprezentaci světa o dané šířce a výšce.
 - Data o světě jsou předána po řádkách v jednorozměrném poli znaků.
 - Interní reprezentace je čistě na vás.
 - `char uzemi(int x, int y)`
 - Metoda vrátí typ území (znak), který se nachází na pozici [x; y].
 - Pokud jsou dané souřadnice mimo rozsah světa, vrátí se hodnota 0 (nikoliv znak '0').
 - `void vykresli(GUI gui)`
 - Pomocí metody `gui.zapis(x, y, znak)` vykreslí na plátno mapu.
 - Předpokládejte, že plátno má dostatečnou velikost a svět vykreslujete od levého horního rohu ([0; 0]).
- Třída `Postava`
 - Reprezentuje postavu.
 - Ta je dána svou pozicí ve světě (se složkami `x` a `y`) a místem, kde má cíl (se složkami `x` a `y`).
 - Veřejné metody
 - `Postava(Svet svet, int startX, int startY, int domovX, int domovY)`
 - Vytvoří postavu a nastaví její výchozí pozici na [startX; startY] a pozici cíle na [domovX; domovY].
 - `boolean jdi(Smer smer)`
 - Pokusí se posunout postavu v daném směru.
 - Vrací `true`, pokud se podařilo daným směrem jít (daným směrem byl volný prostor).
 - V opačném případě vrátí `false` (daným směrem byl znak '#' nebo území mimo svět).
 - `boolean jeDoma()`
 - Metoda vrátí hodnotu `true`, pokud se postava nachází na stejném místě, jako je její cíl.
 - `void vykresli(GUI gui)`
 - Pomocí metody `gui.zapis(x, y, znak)` vykreslí na plátno znak '@' na aktuální pozici postavy v mapě a znak '^' na pozici cíle.
- Třída `Ppa1u08`
 - Spouštěcí třída. Přímo v programu je vytvořen nový svět, např.:

```
Svet svet = new Svet(7, 5, new char[] {
    '#', '#', '#', '#', '#', '#', '#',
    '#', ' ', '#', ' ', ' ', ' ', '#',
    '#', ' ', '#', ' ', ' ', ' ', '#',
    '#', ' ', ' ', ' ', ' ', '#', '#',
    '#', '#', '#', '#', '#', '#', '#',
});
```

- Pozn.: Inicializace pole může být rozdělena do více řádek. Můžete tak snadno dostat náhled na budoucí mapu (viz výše).
- Veřejné metody
 - `static boolean spust(GUI gui, Svet svet, Postava postava)`
 - Metoda spustí nekonečnou smyčku, ve které smaže plátno GUI, vykreslí svět a postavu, a následně zobrazí GUI na `System.out`.
 - Poté načte požadovanou akci a pokusí se jít požadovaným směrem.
 - Pokud je požadovaný směr chybný, metoda vrátí hodnotu `false`.
 - V okamžiku, kdy postava je doma, metoda vrátí hodnotu `true`.
 - `static void main(String[] args)`
 - Vstupní metoda programu.
 - V metodě se vytvoří potřebné instance (tříd `GUI`, `Svet`, `Postava`) a spustí se metoda `spust()`.
 - Pro jednoduchost zvolte shodné rozměry pro `GUI` a `Svet`.
 - Na základě návratové hodnoty metody `spust()` vypíše program buď `"Konecne doma..."`, pokud postava došla domů, nebo `"Asi jsem se ztratil..."`, pokud se cestou něco přihodilo.
- *Hlavně pro informatiky: Všimněte si, že žádné třídy vyjma třídy `GUI` (a výpisu v metodě `main()`) nepracují s konkrétním vstupem či výstupem. To umožňuje oddělit grafickou reprezentaci světa od modelu světa, což má mnoho výhod. Kromě snadné změny způsobu vykreslování například i možnost snadno realizovat jednotkové testy.*

Příklady

- Pro příklad je použit svět definovaný výše a postava začínající na pozici [2; 3] s cílem na [3; 2].
- Návrat domů
 - Uživatel úspěšně dovede postavu domů:

```
#####
# # #
# #^ #
# @ # #
#####
kam dal? 6
```

```
#####
# # #
# #^ #
# @# #
#####
kam dal? 8
```

```
#####
# # #
# #@ #
# # #
#####
Konecne doma...
```

- Ztracený 1

- Uživatel zvolil špatnou akci (číslo 5 neodpovídá žádnému směru):

```
#####
# # #
# #^ #
# @ # #
#####
kam dal? 5
```

Asi jsem se ztratil...

- Ztracený 2

- Uživatel zvolil špatnou akci.
- Číslo 6 sice udává směr na východ, tímto směrem se ale už nelze dál pohybovat.
- Stejný problém by nastal i v okamžiku, kdy by se takto postava dostala mimo hranice světa (např. souřadnice [7; 3])

```
#####
# # #
# #^ #
# @ # #
#####
kam dal? 6
```

```
#####
# # #
# #^ #
# @# #
#####
kam dal? 6
```

Asi jsem se ztratil...