

Domácí úloha 07 (9.11. až 20.11.2022 – 4 body) – ppa1u07.jar

- Podle následujícího popisu naimplementujte jednoduchý soubojový systém pro RPG (RPG = Role Playing Game: hra na hrdiny, počítačová, stolní či v reálném světě).
- Dodržte názvy tříd, hlavičky a funkcionalitu veřejných metod dle následujícího popisu.
- Se třídami lze manipulovat pouze pomocí veřejných metod uvedených v zadání.
- Případné privátní atributy (proměnné), metody či další třídy vytvořte dle uvážení.
- Pomocí dokumentačních komentářů úlohu řádně okomentujte.
 - o Komentář by měl být u každé veřejné třídy, metody i atributu (třídy i instance)
 - o Pro třídy uveďte i autora a verzi (jako verzi uveďte aktuální datum ve tvaru RRMMDD).
- Výčet `Ruka`
 - o Výčtový typ `Ruka` má pouze položky `LEVA` a `PRAVA`.
- Třída `Postava`
 - o `Postava` ve hře je určena svým jménem, silou (základní útočná síla), hbitostí (základní obranná síla) a vitalitou (počáteční zdraví).
 - o Každá postava navíc může v levé a pravé ruce držet zbraň, která ovlivňuje celkovou obrannou a útočnou sílu postavy.
 - o Veřejné metody
 - `Postava(String jmeno, int sila, int hbitost, int vitalita)`
 - Konstruktor vytvoří instanci postavy se jménem, základní silou, hbitostí a vitalitou.
 - `boolean vezmiZbran(Ruka ruka, Zbran zbran)`
 - Pokud postava nemá v ruce `ruka` zbraň, vezme ji a vrátí hodnotu `true`.
 - V opačném případě se nic nestane a vrátí hodnotu `false`.
 - `int branSe(int utok)`
 - Sníží zdraví postavy v závislosti na síle útoku a aktuální obranné síle (tj. součet hbitosti postavy a obranné síly držených zbraní).
 - Aktuální zdraví se zmenší o rozdíl `utok` a aktuální obrana.
 - V případě slabého útoku (útok je menší než obrana) se zdraví nemění.
 - Metoda vrátí množství ubraného zdraví.
 - `int zautoc()`
 - Vrátí celkovou útočnou sílu (tj. součet síly postavy a útočné síly držených zbraní).
 - `boolean jeZiva()`
 - Pokud je aktuální zdraví větší než 0, postava je živá (tj. metoda vrátí `true`).
 - `String toString()`
 - Vrátí informace o postavě ve formátu:
`jmeno [aktuální zdraví/vitalita] (celková útočná síla/celková obranná síla)`
- Třída `Zbran`
 - o Zbraň je ve hře určena svým názvem a útočnou a obrannou silou.

- Veřejné metody
 - `Zbran(String nazev, int utok, int obrana)`
 - Konstruktor vytvoří zbraň s názvem a silami.
 - `int getUtok()`
 - Getter vrátí útočnou sílu zbraně.
 - `int getObrana()`
 - Getter vrátí obrannou sílu zbraně.
 - `String toString()`
 - Vrátí informace o zbrani ve formátu:

`nazev (útočná/obránná síla)`

- Třída `Ppau07`

- Hlavní třída, která načte z klávesnice informace o první postavě, zbrani do levé a zbrani do pravé ruky. Následně načte stejné informace o druhé postavě.
- Poté spustí souboj mezi postavami (1. postava útočí, 2. postava se brání a opačně) dokud některé z postav neklesne zdraví pod 1.
- Na konci musí aplikace vypsat informace o vítězi.
 - Viz `Postava.toString()`.
- Veřejné metody
 - `static Postava nactiPostavu(Scanner sc)`
 - Z předaného `Scanneru` načte postupně jméno, sílu, hbitost a zdraví a vytvoří novou postavu.
 - Každý údaj je na samostatné řádce.
 - Pro načítání dat používejte metodu `Scanneru` `nextLine()` a získanou hodnotu převeďte na odpovídající datový typ
 - Např. `Integer.parseInt(sc.nextLine())`;
 - `static Zbran nactiZbran(Scanner sc)`
 - Z předaného `Scanneru` načte postupně název, útočnou a obrannou sílu a vytvoří novou zbraň.
 - Pokud se zadá prázdný název, metoda okamžitě vrátí hodnotu `null` (zbytek informací nenačítá).
 - Každý údaj je na samostatné řádce.
 - Pro načítání dat používejte metodu `Scanneru` `nextLine()` a získanou hodnotu převeďte na odpovídající datový typ
 - Např. `Integer.parseInt(sc.nextLine())`;
 - `static void vyzbrojPostavu(Postava postava, Zbran leva, Zbran prava)`
 - Vyzbrojí postavu dodanými zbraněmi.
 - Pokud již postava nějakou zbraň měla, zůstane jí
 - Viz `Postava.vezmiZbran()`.
 - Obecně zbraň může být `null`.
 - `static Postava souboj(Postava postava1, Postava postava2)`
 - Spustí souboj mezi postavami.
 - Souboj končí v okamžiku, kdy aktuální zdraví některé z postav klesne pod 1.

- Útočit začíná postava1.
- Útok probíhá tak, že se zjistí síla útoku útočící postavy (viz `Postava.zautoc()`) a bráníci se postava se pokusí tomuto útoku ubránit (viz `Postava.branSe()`).
- Metoda vrátí přeživší postavu.
- `static void main(String[] args)`
 - Vstupní bod programu.
 - Načte první postavu a zbraň pro levou a pravou ruku.
 - Načte druhou postavu a zbraň pro levou a pravou ruku.
 - Obě postavy vyzbrojí a spustí souboj, ve kterém začíná útočit první postava.
 - Po ukončení souboje vypíše informace o vítězi ve formátu:

Vitez: + `postava.toString()`

Příklad

- Vstup

- Pro testování je vhodné připravit si soubory pro vstup a použít přesměrování vstupu.
- Příklad vstupního souboru (prázdná řádka znamená chybějící zbraň):

```
Thorygg
5
3
80

Tezky mec
15
2
vlk
10
10
30
```

- Výstup

- Nutný minimální výstup je řádka s oznámením vítěze a jeho aktuálním stavem:

Vitez: Thorygg [70/80] (20/5)

- Pro účely ladění je vhodnější nechat si vypsat i průběh souboje. Tyto výpisy mohou pro odevzdání klidně zůstat (důležitá je pouze řádka s vítězem). Např.:

Souboj: Thorygg [80/80] (20/5) vs vlk [30/30] (10/10)

```
utoci Thorygg [80/80] (20/5) a dava 10 zraneni
utoci vlk [20/30] (10/10) a dava 5 zraneni
utoci Thorygg [75/80] (20/5) a dava 10 zraneni
utoci vlk [10/30] (10/10) a dava 5 zraneni
utoci Thorygg [70/80] (20/5) a dava 10 zraneni
```

Vitez: Thorygg [70/80] (20/5)