

Domácí úkol 2

Termín odevzdání:	15.03.2020 23:59:59
Hodnocení:	5.0000
Max. hodnocení:	5.0000 (bez bonusů)
Odevzdaná řešení:	1 / 20
Nápovědy:	0 / 10

V tomto úkolu implementujeme samotný hrací plán a dlaždice na něm. Postupujte přesně podle následujícího zadání.

Pravidla hry, které budete v průběhu jednotlivých úkolů implementovat, najdete **zde**.

Zadání úkolu

- Pokračujte v Java projektu, který jste vytvořili v předchozím úkolu.
- Uvnitř balíčku thedrake ve vašem projektu vytvořte výčtový typ PlayingSide s hodnotami ORANGE a BLUE, který představuje barvu, za kterou hráč hraje.
- Dále vytvořte rozhraní Tile. To představuje dlaždici na hracím plánu. Každá dlaždice umí zodpovědět, zda se na ni dá vstoupit nebo zda obsahuje jednotku.

```
// Vrací True, pokud je tato dlaždice volná a lze na ni vstoupit.
public boolean canStepOn();
```

```
// Vrací True, pokud tato dlaždice obsahuje jednotku
public boolean hasTroop();
```

- Vytvořte implementaci rozhraní Tile jménem TroopTile. Tato immutable třída bude představovat dlaždici, na které stojí jednotka. Tato dlaždice si pamatuje jednotku, která na ní stojí, barvu, za kterou jednotka hraje, a stranu, na kterou je jednotka otočena (rub nebo líc). Pozor tedy na to, že náš návrh funguje tak, že barvu a stranu si nepamatuje jednotka sama, ale až dlaždice, na které stojí. To nám v budoucnu ušetří hodně práce.

```
// Konstruktor
public TroopTile(Troop troop, PlayingSide side, TroopFace face)
```

```
// Vrací barvu, za kterou hraje jednotka na této dlaždici
public PlayingSide side()
```

```
// Vrací stranu, na kterou je jednotka otočena
public TroopFace face()
```

```
// Jednotka, která stojí na této dlaždici
public Troop troop()
```

```
// Vrací False, protože na dlaždici s jednotkou se nedá vstoupit
public boolean canStepOn()
```

```
// Vrací True
public boolean hasTroop()
```

```
// Vytvoří novou dlaždici, s jednotkou otočenou na opačnou stranu
// (z rubu na líc nebo z líce na rub)
public TroopTile flipped()
```

- Z přípravy (stáhněte si níže "Vzorová data") si vezměte rozhraní BoardTile, které představuje dlaždice, které neobsahují jednotky. To jsou prázdné dlaždice nebo hory. Prohlédněte si jeho implementaci a vložte ho do vašeho balíčku.
- Dále si z přípravy vezměte rozhraní TilePos, které představuje pozici dlaždice na hracím plánu. Souřadnice dlaždice se interně reprezentují jako dvě celá čísla i (číslo sloupce) a j (číslo řádku). Dlaždici v levém dolním rohu odpovídá souřadnice i=0, j=0. Pozici dlaždice je pak možné reprezentovat také pomocí lidsky vstřícnějších souřadnic typu a1, c4 apod. viz **pravidla hry**. Rozhraní TilePos má tedy převodní metody pro práci s oběma typy souřadnic. Dále obsahuje přetíženou metodu step(), která vytvoří novou souřadnici posunutou ve směru zadaného offsetu. Metoda stepByPlayingSide() navíc dělá to, že posune tuto souřadnici podle barvy hráče. Pokud hraje modrý, funguje jako metoda step(). Pokud hraje oranžový, udělá posun s obrácenou y souřadnicí. Metoda isNextTo() vrátí true, pokud tato pozice sousedí se zadanou pozicí. Všimněte si implementace OFF_BOARD, která představuje pozici, která se nachází mimo hrací desku. Vždy, když by nějaká z metod step... měla vykročit mimo desku, vrací právě pozici OFF_BOARD.
- V přípravě najdete také třídu BoardPos implementující rozhraní TilePos;. Dále zde najdete třídu PositionFactory; která se stará o vytváření pozic na hracím plánu zadané velikosti. Díky tomu mohou samy pozice kontrolovat, jestli se náhodou nenacházejí mimo hranice hrací plochy.
- Výzva tohoto úkolu je vytvořit třídu Board, představující hrací plán. Ten sestává z dvourozměrného pole dlaždic typu BoardTile. Vzhledem k tomu, že design celé hry je immutable, i třída Board je immutable. Chceme-li tedy změnit nějakou dlaždici na jinou, musíme vytvořit novou kopii této třídy s touto novou dlaždici na správném místě. Dlaždice budeme uchovávat ve dvourozměrném poli, které tedy bude potřeba vždy celé zkopírovat. Zde se může hodit metoda Object.clone(), která umí naklonovat **jednorozměrné pole**.
- Třída Board je lehce komplikovanější, stáhněte si šablonu pro její implementaci z přípravy. Šablona obsahuje jednu vnitřní třídu Board.TileAt. Je to pomůcka, pomocí které říkáme na jaké pozici je jaká dlaždice, protože dlaždice si samy svoji pozici nepamatují. Toto se nám hodí pro metodu withTiles viz níže.
- V šabloně třídy Board implementujte vyznačené metody

```
// Konstruktor. Vytvoří čtvercovou hrací desku zadaného rozměru, kde všechny dlaždice jsou prázdné, tedy BoardTile.EMPTY
public Board(int dimension)
```

```
// Rozměr hrací desky
public int dimension()
```

```
// Vrací dlaždici na zvolené pozici.
public BoardTile at(TilePos pos)
```

```
// Vytváří novou hrací desku s novými dlaždicemi. Všechny ostatní dlaždice zůstávají stejné
public Board withTiles(Board.TileAt ...ats)
```

```
// Vytvoří instanci PositionFactory pro výrobu pozic na tomto hracím plánu
public PositionFactory positionFactory()
```

- Overte, že váš kód splňuje testy (stáhněte si níže "Vzorová data"). Spuštějte vždy třídu MainSuite.java.

Odevzdání úkolu

- Vytvořte soubor thedrake.zip s přesně touto strukturou složek a souborů

```
/thedrake
/thedrake/Board.java
/thedrake/BoardPos.java
/thedrake/BoardTile.java
/thedrake/Offset2D.java
/thedrake/PlayingSide.java
/thedrake/PositionFactory.java
/thedrake/Tile.java
/thedrake/TilePos.java
/thedrake/Troop.java
/thedrake/TroopFace.java
/thedrake/TroopTile.java
```

- Než budete úkol posílat, seznamte se s pravidly pro odevzdávání úkolů (viz. Moodle). Najdete zde důležité informace nezbytné k tomu, aby byl váš úkol přijat.

Vzorová data:	Download
----------------------	-----------------

1	03.03.2020 11:20:28	Download
----------	----------------------------	-----------------

Stav odevzdání:	Ohodnoceno
Hodnocení:	5.0000

- Hodnotitel: automat**
 - Program zkompilován
 - Test 'Standard test': Úspěch
 - Dosaženo: 100.00 %, požadováno: 100.00 %
 - Úspěch v závazném testu, hodnocení: 100.00 %
 - Celkové hodnocení: 100.00 %
- Celkové procentní hodnocení: 100.00 %
- Celkem bodů: 1.00 * 5.00 = 5.00