java-7-nodarbiba-masivi-2-uzdevums.md

Uzdevuma apraksts

Tēma 7 - Masīvi.

Uzdevums 2. Neobligāts

a. Kick-Boxing svara kategoriju tabula

- dota sekojoša Kick-Boxing svaru tabula.

weight class	from	to
Fly Weight	0	112
Super Fly Weight	112	115
Bantam Weight",	115	118
Super Bantam Weight	118	122
Feather Weight	122	126
Super Feather Weight	126	130
Light Weight	130	135
Super Light Weight	135	140
Welter Weight	140	147
Super Welter Weight	147	154
Middle Weight	154	160
Super Middle Weight	160	167
Light Heavy Weight	167	174
Super Light Heavy Weight	174	183
Cruiser Weight	183	189

```
Super Cruiser Weight       189   198
Heavy Weight               198   209
Super Heavy Weight         209
```

- izveidojam jaunu pakotni vārds_uzvārds.lesson7.task_2;
- izveidojam klasi WeightCategoy
- Lai saglabātu svaru tabulu, izveidojam 2 masīvus (konstantes), klases līmenī (lauki vai instances mainīgie)
    - i. String masīvs kurš glabā Svaru Kategorijas nosaukumus ("Fly Weight ", "Super Fly Weight")

```
public final String[] WEIGHT_CATEGORY = new String[]{"Fly Weight ", ...};
```

- ii. Int (veselu skaitļu) masīvs kurš glabā vecumu robežu. (0,112,115,..)

```
public int[] WEIGHT_INTERVAL = new int[]{0,112,115, ...};
```

- izveidot metodi kura atgriež svaru kategorijas nosaukumu, ja dots svars (int skaitlis).
    - metodes implementācijai izmantot ciklu!
    - metodi nodefinējam kā statisku.

```
public static String getWeightCategoryName(int weight){
//    ....
}
```

- izveidojam testa klasi šīs metodes pārbaudei (analoģiski 1. uzdevumam)
    - right click -> Generate -> Test -> JUnit 4, iezīmējiet interešejošo metodi getWeightCategoryName
    - pievienojiet testa metodei assert() izsaukumu
    - assert metode salīdzina sagaidāmo un aktuālo vērtību!

```
    @Test
    void min() {
        WeightCategory app = new WeightCategory();

        int[] arrayInt = new int[]{2,9,100,1};
        int expectedResult = "Super Bantam Weight";
        String actualResult = app.app.getWeightCategoryName(120);
        assertEquals(expectedResult,actualResult,"should return expected result");
```

- Palaižam testu. Līdzīgi kā palaižot main metodi (uzklikšķinot uz zilā stūrīša pie klases)
- Kreisajā pusēe apakšā redzams testa rezultāts (zaļš ir veiksmīgs, sarkans ir neveiksmīgs)

b. Kāršu sajaukšana un meklēšana.

Apskats:

- Turpinām ar kāršu spēli.
  - Izveidosim kāršu kopu (card deck)
  - sajauksim kārtis
  - noskaidrosim cik ātri varam atrast viszaz vienu katras šlakas kārti.
- Izmantosim Nodarbibas 5 1. uzdevuma Card klasi.
- Ar copy/Paste iekopējam Card klasi iekš pakotnes vārds_uzvārds.lesson7.task_2;
- Izlabojam sintakses kļudas
- Izveidojam CardShuffle klasi šajā pašā pakotnē.
- izveidojam metodi void shuffleCards(Card[] deck){}
- ši metode COUNT_SHUFFLE (konstante mūsu klasē) reizes samaina kārti 2 random pozīcijās izmantojot Math.Random klasi.

```
void shuffleCards(Card[] deck){
       // Loop constant number of times
       Random rnd = new Random();
```

```
        int pos1 = rnd.nextInt(deck.length()-1);
        int pos2 = rnd.nextIntdeck.length()-1);
        // switch cards at 2 random positions
        Card card1 = deck[pos1];
        deck[pos1]=deck[pos2];
        deck[pos2]=card1;
        // ...
}
```

- Uzrakstīt metodi, kura atgriež skaitli kurš pasaka pēc cik reizēm ir atrastas visu šlaku kārtis.

```
public int takeAllSuitsAtLeastOnce(Card[] deck){
// your implementation here
}
```

- tālāk turpinam ar main() metodi. Lai izsauktu augstāk nodefinētās metodes, apzīmējam tās ar static.
- Izveidojam pilnu Card Deck (Kāršu kopu). Kopā 4 * 13 kāršu.
    - Izveidojam masīvu CardDek

```
Card[] cardDeck = new Card[4*13];
```

- ar iekļautu ciklu palīdzību noinicializējam cardDeck masīvu.

```
int cardIdx = 0;
for(int cSuite=0; cSuite<4; cSuite++){
  for(int cValue=0; cValue<13; cValue++){
     cardDeck[cardIdx++] = new Card(cValue,cSuite);
  }
}
```

- izsaucam metodi shuffleCards(Card[] deck) lai sajauktu kāršu kopu
- izsaucam metodi takeAllSuitsAtLeastOnce(Card[] deck) un izdrukājam rezultātu

- palaižam programmu vairākas reizes. Vai kārtis ir sajauktas labi?