

Tēma 5




































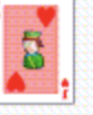















Uzdevums 1 - Objektu salīdzināšana un toString izmantošana

Kopsavilkums

Izveidosim Spēļu Kārts modeli.

- Kārts modelis ir tabula kurai pa rindām ir šlaka un pa vertikāli punktu skaits.
- Izveidosim kārti kas bāzēta uz šiem diviem parametriem.
- Konkrētu kārti varēsīm parādīt teksta formā un pārbaudīt vai cita kārts nesakrīt ar doto.
- zemāk skatīt kāršu komplektu, šis ir modelis uzdevuma veikšanai.
- pievienosim privātu bezargumentu konstruktoru lai klases objektu varētu izveidot tikai norādot 2 argumentus kuri raksturo konkrēto kārti.

Example set of 52 poker playing cards

Suit	Ace	2	3	4	5	6	7	8	9	10	Jack	Queen	King
Clubs													
Diamonds													
Hearts													
Spades													

Soļi

- Izveidot Card klasi
 - Card klases konstruktors ir ar 2 parametriem, publisks
 - Bezargumentu konstruktors privāts.

```
public class Card{ // klase
```

```
// card statee
```

```
int theValue;
```

```
int theSuit;
```

```

// card deck configuration
// theSuit values
public final static int SPADES = 0;    // Pīķis
public final static int HEARTS = 1;    // Ercens
public final static int DIAMONDS = 2;  // Kārvs
public final static int CLUBS = 3;     // Kreicis
public final static int JOKER = 4;     //

// theSuit values
public final static int ACE = 1;        // Dūzis
public final static int CARD_2 = 2;     // Kārts
public final static int CARD_3 = 3;     // Kārts
public final static int CARD_4 = 4;     // Kārts
public final static int CARD_5 = 5;     // Kārts
public final static int CARD_6 = 6;     // Kārts
public final static int CARD_7 = 7;     // Kārts
public final static int CARD_8 = 8;     // Kārts
public final static int CARD_9 = 9;     // Kārts
public final static int CARD_10 = 10;   // Kārts
public final static int JACK = 11;      // Kalps
public final static int QUEEN = 12;     // Karaliene
public final static int KING = 13;      // Karalis

public Card(int theValue, int theSuit) { // constructor
    // code here
}

.....

}

```

- pielikt getterus, setterus theSuit un theValue instances mainīgajiem
 - setterus / getterus pielikt kā protected (public vietā).
 - Izmantojam Intellij enkapsulācijas helperi:

- Right Click -> Refactor -> Encapsulation
- Dialogā atzīmējam vajadzīgos laukus
- pievienot metodi getSuiteTitle() kura atgriež kārts šlaku

izmantojam switch() zarošanās konstrukciju:

```
String getCardSuiteTitle(){  
    switch(this.theValue){  
        ACE: return "ACE";  
        CARD_2 : return "2";  
        // ...  
        default: return null; // nav iespējams ja korekts funkcijas arguments ir nodots!  
    }  
  
}  
  
}
```

- pievienot metodi getValueTitle() kura atgriež kārts lielumu
 - izmantojot switch() zarošanās konstrukciju dekodējam kārts lielumu:
 - theValue -> "value text"
 - rīkojamies analogiski kā getCardSuiteTitle() gadījumā.
- Aizvietot noklusēto equals metodi vienādas kārts atrašanai.
 - redaktorā brīvā vietā ārpus metodēm un iekšpusē klasē sākam rakstīt: equals, kamēr parādās Intellij saraksts kurā redzams equals(), Spiežam ENTER.
 - mērķis ir lai Cards klases objektus salīdzinātu pēc theValue un theSuit lauku vērtībām.

- Apskatiet ievietoto kodu vai tas apmierina kāršu objektu salīdzināšanas kritērijus.
- Aizvietot toString() metodi.
 - redaktorā brīvā vietā ārpus metodēm un iekšpusē klasē sākam rakstīt: toString, kamēr parādās Intellij saraksts kurā redzams toString(), Spiežam ENTER.
 - toString metode ir bez argumentiem un atgriež String tipu
 - Gribam atgriezt šādu tekstu "DIAMOND 2" vai "HEART 4" vai "CLUBS Joker"
- (Ne Obligāti) Aizvietot comapareTo() metodi kāršu vēertības salīdzināšanai

compareTo() atgriež:

0 = objekti vienādi

1 = lielāks

2 = mazāks

- (Ne Obligāti) salīdzināšanas likumi:
 - salīdzināšana balstīta uz theValue ja theSuit ir vienāds
 - ja theSuit ir dažādi, tad SPADES > HEARTS > DIAMONDS > CLUBS
 - redaktorā brīvā vietā ārpus metodēm un iekšpusē klasē sākam rakstīt: compareto, kamēr parādās Intellij saraksts kurā redzams equals(), Spiežam ENTER.
 - Intellij noģenerēs gan compareTo gan hashCode() metožu aizvietotājus.
 - pārbaudam izveidoto kodu un papildinām ar sekojošo

```
// note that class can have
```

```
// class Type implements Comparable<Type>
```

```

@Override
public int compareTo(Card other) {
    // salīdzināšana balstīta uz theValue ja theSuit ir vienāds
    // ja theSuit ir dažādi, tad
    // SPADES > HEARTS > DIAMONDS > CLUBS
    // atgriež +1 ja lielāks
    // = 0 , ja vienāds
    // < 0 , ja mazākss
    if(this.theValue == other.theValue && this.theSuit == other.theSuit) return 0;
    else if(this.theSuit >= other.theSuit && this.theValue > other.theValue) return 1;
    else return -1;
}

```

- Izveidojam jaunu klasi CardApp ievietojot to apakšpakotnē cardapp (pakotnju nosaukumiem jābūt lowercase burtiem). t.i ja pakotne kurā atrodas card ir homework_5.task1 tad CardApp ievietojams homework_5.task1.cardapp
- CardApp main metode:
 - izveido 2 vienādas Cards objektus (ar theSuit un theValue) un parbauda vai tie vienādi ar equals metodi.

```

Card card1 = new Card(DIAMONDS,CARD_2);
Card card2 = new Card(DIAMONDS,CARD_2);
System.out.println(" cards are equal ? "+card1.equals(card2));

```

- izveido 2 dažādus Cards objektus (ar theSuit un theValue) un pārbauda kura kārts lielāka ar compareTo metodi.

```

Card card1 = new Card(DIAMONDS,CARD_2);
Card card2 = new Card(DIAMONDS,CARD_5);
System.out.println(" which cards is greater ? "+(card1.compareTo(card2)); // =0 equals, 1=greater, 2 = Smaller

```

- izdrukā kārts nosaukumu t.i. `println(theCard)` parādot kārts nosaukumu norādot `Card` klases objektu.

```
Card card1 = new Card(DIAMONDS,CARD_2);
Card card2 = new Card(DIAMONDS,QUEEN);
// tagad varam gan šādi
System.out.println(" card 1 = "+card1.getValueTitle()+" "+card1.getSuiteTitle());
// gan ievietojot objektu, pateicoties toString() aizvietojumam
System.out.println(" card 1 = "+card1+"card 2 = "+card2);
```