ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ
Одсек за СИ

Београд 22.03.2019.

Први колоквијум из
Објектно оријентисаног програмирања II

1) *(укупно 70 поена)* Саставити на језику *Java* следећи пакет класа:

- *(15 поена)* ***Број*** телефона се ствара задавањем текстуалног кода државе, позивног броја, броја телефона (на пример „381“, „11“, „1234567“) и индикатора да ли се ради о фиксном телефону. Могуће је проверити да ли се број телефона налази у истој држави као други задати број телефона. Индикатор који говори да ли се ради о фиксном телефону је могуће дохватити. Текстуални опис броја телефона је облика `+kod_drzave pozivni_broj broj_telefona`.
- *(30 поена)* ***Услуга*** се ствара задавањем броја телефона који је иницирао услугу и броја телефона коме је услуга упућена. Могуће је дохватити целобројну цену услуге. Текстуални опис услуге је облика `broj_od -> broj_ka`.
- ***Позив*** је услуга за коју се додатно задаје и трајање у секундама. Цена позива се формира на основу цене успостављања везе и цене започетог минута разговора. Успостављање везе у домаћем саобраћају је бесплатно, док се сваки започети минут наплаћује 10 динара. За позиве ка иностранству успостављање везе се наплаћује 30 динара, а сваки започети минут 50 динара. Трајање позива 0 секунди означава да веза није успостављена, те је цена таквог позива 0 динара. Текстуални опис позива додатно садржи и трајање позива у формату `(minuti:sekunde)`.
- ***Порука*** је услуга за коју се додатно задаје текст поруке. Могуће је проверити да ли је порука послата. Поруку није могуће послати са броја фиксног телефона или на број фиксног телефона. Цена послате поруке у домаћем саобраћају је 3 динара, а ка иностранству 20 динара. За поруку коју није могуће послати цена је 0 динара. Текстуални опис поруке додатно садржи и текст поруке.
- *(25 поена)* ***Корисник*** се ствара задавањем броја телефона и почетног броја услуга које је за корисника потребно памтити. Број услуга које се памте се увећава по потреби за по 10. Могуће је за корисника евидентирати позив, при чему се задаје број телефона који је позван и трајање позива, или поруку, при чему се задаје број телефона на који је порука послата, као и текст поруке. Повратна вредност обе операције је цена услуге. Могуће је дохватити укупну цену свих извршених услуга. Текстуални опис корисника се састоји од његовог броја телефона у првом реду, а затим свих извршених услуга у хронолошком редоследу, при чему је свака услуга у засебном реду.

(0 поена) Приложена је главна функција која направи једног корисника са задатим бројем телефона, затим евидентира неколико позива и порука и на крају испише текстуални опис корисника и цену свих извршених услуга на стандардни излаз.

НАПОМЕНЕ:

а) Колоквијум траје **100** минута.
б) Називе типова ускладити са називима апстракција из текста задатка, али користити латинично писмо и велико почетно слово.
в) Рад се предаје искључиво на предвиђеном мрежном диску.
г) На располагању је приступ *Web* адреси: https://docs.oracle.com/en/java/javase/11/ .
д) Није дозвољено уз себе имати електронске уређаје, без обзира да ли су укључени или искључени.
ђ) Резултати колоквијума биће објављени на *Web*-у на адреси: http://rti.etf.rs/rti/ir2oo2/index.html

```java
package main;

import telefon.Broj;
import telefon.Korisnik;
import telefon.Poruka;
import telefon.Poziv;

public class Main {
   public static void main(String[] args) {
      Broj b1 = new Broj("381", "64", "0000000", false);
      Broj b2 = new Broj("381", "11", "0000000", true);
      Broj b3 = new Broj("454", "59", "0000000", false);

      Poziv poziv = new Poziv(b1, b3, 253);
      Poruka poruka = new Poruka(b1, b3, "Dobar dan");

      System.out.println(poziv);
      System.out.println(poruka);

      System.out.println();

      Korisnik k = new Korisnik(b1, 2);
      k.evidentirajPoruku(b3, "Zdravo");
      k.evidentirajPoruku(b3, "Cao");
      k.evidentirajPoziv(b2, 74);

      System.out.println(k);
      System.out.println("Ukupna cena: " + k.ukupnaCena());
   }
}
```

Пример извршавања програма:

```
+381 64 0000000 -> +454 59 0000000 (4:13)
+381 64 0000000 -> +454 59 0000000 Dobar dan

+381 64 0000000
+381 64 0000000 -> +454 59 0000000 Zdravo
+381 64 0000000 -> +454 59 0000000 Cao
+381 64 0000000 -> +381 11 0000000 (1:14)

Ukupna cena: 60
```

```java
package telefon;
public class Broj {
   private String kod, pozivni, broj;
   private boolean fiksni;

   public Broj(String kod, String pozivni,
            String broj, boolean fiksni){
      this.kod = kod;
      this.pozivni = pozivni;
      this.broj = broj;
      this.fiksni = fiksni;
   }
   public boolean istaDrzava(Broj broj){
      return kod.equals(broj.kod);
   }
   public boolean fiksni(){
      return fiksni;
   }
   public String toString(){
      return "+"+kod+" "+pozivni+" "+broj;
   }
}
```

```java
public abstract class Usluga {
   protected Broj sa, na;

   public Usluga(Broj sa, Broj na){
      this.sa = sa;
      this.na = na;
   }
   public abstract int cena();
   public String toString(){
      return sa+" -> "+na;
   }
}
```

```java
public class Poziv extends Usluga {
   private int trajanje;

   public Poziv(Broj sa, Broj na,
            int trajanje) {
      super(sa, na);
      this.trajanje = trajanje;
   }
   @Override
   public int cena() {
      int minuta = (trajanje+59)/60;
      if(minuta == 0) return 0;
      if(sa.istaDrzava(na))
         return minuta*10;
      else
         return 30+minuta*50;
   }
   @Override
   public String toString() {
      return super.toString() + " ("
       + (trajanje/60)+":"+(trajanje%60)+")";
   }
}
```

```java
public class Poruka extends Usluga {
   private String tekst;

   public Poruka(Broj sa, Broj na,
            String tekst) {
      super(sa, na);
      this.tekst = tekst;
   }
   public boolean poslata(){
      return !sa.fiksni() && !na.fiksni();
   }
   @Override
   public int cena() {
      if(!poslata()) return 0;
      if(sa.istaDrzava(na)) return 3;
      else return 20;
   }
   @Override
   public String toString() {
      return super.toString()+ " "+tekst;
   }
}
```

```java
public class Korisnik {
   private Broj broj;
   private Usluga usluge[];
   private int trenutno;

   public Korisnik(Broj broj, int brUsluga) {
      this.broj = broj;
      this.usluge = new Usluga[brUsluga];
   }
   private void dodajUslugu(Usluga usluga){
      if(trenutno == usluge.length){
         Usluga u[] = new Usluga[trenutno+10];
         for(int i=0;i<trenutno;i++){
            u[i] = usluge[i];
         }
         usluge = u;
      }
      usluge[trenutno++] = usluga;
   }
   public int evidentirajPoziv(Broj broj,
            int trajanje){
      Poziv p =
         new Poziv(this.broj, broj, trajanje);
      dodajUslugu(p);
      return p.cena();
   }
   public int evidentirajPoruku(Broj broj,
             String tekst){
      Poruka p =
         new Poruka(this.broj, broj, tekst);
      dodajUslugu(p);
      return p.cena();
   }
   public int ukupnaCena(){
      int cena = 0;
      for(int i=0;i<trenutno;i++)
          cena+=usluge[i].cena();
      return cena;
   }
   @Override
   public String toString() {
      StringBuilder sb = new StringBuilder();
      sb.append(broj);
      sb.append("\n");
      for(int i=0;i<trenutno;i++){
         sb.append(usluge[i]);
         sb.append("\n");
      }
      return sb.toString();
   }
}
```