ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ
Одсек за СИ Београд 27.04.2018.

Други колоквијум из Објектно оријентисаног програмирања II

Написати на језику *Java* следећи пакет типова:

- ***Скуп*** се састоји од прозвољног броја података целобројног типа. Ствара се празан након чега се подаци додају појединачно. Додавање податка који се већ налази у скупу се игнорише, а повратна вредност приликом додавања одређује успешност операције. Може се проверити да ли се податак већ налази у скупу. Скуп је могуће испразнити. Текстуални опис је облика **{***елем***,** *елем***,** *...***}**, где *елем* представља један елемент (податак) скупа.
- ***Генератор*** јединствених случајних бројева генерише на захтев случајан цео број. Приликом стварања генератора задају се горња и доња граница опсега вредности генерисаних бројева. Генератор памти скуп претходно генерисаних бројева и сваки пут генерише број који се не налази у том скупу. Грешка (`GreskaGenerisanje`) је уколико се тражи генерисање новог броја, а сви бројеви из постављеног опсега су већ генерисани. Могуће је испразнити скуп генерисаних бројева.
- Активни ***бубањ*** циклички (са паузом између 0 и 50 ms) генерише случајне бројеве уз помоћ садржаног генератора случајних бројева, у опсегу који се задаје приликом стварања. Бубањ је могуће покренути и трајно зауставити. Приликом покретања бубањ генерише нови број и зауставља се привремено, док тај број не буде узет. Генерисани број је могуће узети, при чему, уколико број још увек није генерисан, онај ко узима број чека док бубањ не генерише нови број. Након једног узимања генерисаног броја потребно је генерисати нови број који би могао да се узме. Могуће је проверити да ли је бубањ већ генерисао нови број.
- Бинго ***листић*** има аутоматски генерисани целобројни индентификатор и може да има задат број целих бројева. Ствара се празан након чега се бројеви појединачно додају. Није потребно проверавати нерегуларне ситуације приликом додавања. Могуће је проверити колико бројева са листића се налази у задатом скупу. Текстуални опис листића је облика **[***ид***]{***број***,** *број***,** *...***}**, где *број* представља један број на листићу.
- Активни ***бинго*** је игра која се састоји од једног бубња, једног генератора бројева за попуњавање листића и задатог броја листића који учествују у извлачењу. Приликом стварања задаје се и опсег бројева које бубањ генерише и који се могу наћи на листићима, као и број бројева који се налази на листићима. Могуће је генерисати нови листић који се том приликом ствара и аутоматски попуњава унапред задатим бројем бројева у задатом опсегу. Задати листић је могуће додати листићима који учествују у бинго извлачењу. Није потребно проверавати грешке приликом додавања листића на списак. Бинго врши извлачење бројева из бубња. Могуће је покренути извлачење. Током извлачења се из бубња узима и исписује на стандардни излаз један по један број све док неки од листића који учествују у извлачењу не постане добитни (сви бројеви са листића су извучени). Грешка (`GreskaGenerisanje`) је ако се покуша генерисање или додавање нових листића током извлачења. По завршетку извлачења могуће је дохватити добитни листић (било који у случају постојања више њих).

Приложена је главна функција која направи једну бинго игру са листићима који имају по 15 бројева у опсегу од 1 до 99, затим затражи генерисање неколико бинго листића које испише и потом покрене извлачење, те испише добитни листић на крају. Грешке пријављивати изузецима типа класа које садрже текст поруке.

НАПОМЕНЕ:

а) Време за израду задатка је **100** минута.

б) Рад се предаје искључиво на предвиђеном мрежном диску.

в) Називе типова ускладити са називима апстракција из текста задатка, али користити латинично писмо и велико почетно слово.

г) На располагању је *Oracle* документација на *Web* адреси: https://docs.oracle.com/javase/8/docs/api/. Није дозвољено имати поред себе друге материјале, нити уз себе имати електронске уређаје, без обзира да ли су укључени или искључени.

д) Резултати колоквијума биће објављени на *Web*-у на адреси:http://rti.etf.bg.ac.rs/rti/ir2oo2/index.html

```java
package test;

import bingo.Bingo;
import bingo.Bubanj;
import bingo.Generator;
import bingo.GreskaGenerisanje;
import bingo.Listic;
import bingo.Skup;

public class Main {
   public static void main(String args[]){

      try {
         Skup s = new Skup();
         if(s.postoji(5)) s.dodaj(5);

         Generator g = new Generator(1,5);

         Bubanj b = new Bubanj(1,39);
         b.pokreni();
         for(int i=0;i<7;i++) {
            int br = b.sledeci();
         }
         b.zaustavi();

         int maxBrListica = 100;
         int minBrNaListicu = 1;
         int maxBrNaLisitcu = 99;
         int brBrojevaNaListicu = 4;

         Bingo bingo = new Bingo(maxBrListica, minBrNaListicu, maxBrNaLisitcu, brBrojevaNaListicu);
         for(int i=0;i<10;i++) {
            Listic l = bingo.novi();
            System.out.println(l);
            bingo.dodajListic(l);
         }
         bingo.pokreni();
         bingo.join();
         System.out.println();
         System.out.println("Dobitni listic: " + bingo.dobitni());

      } catch (GreskaGenerisanje e) {
         System.out.println(e.toString());
      } catch (InterruptedException e) {}
   }
}
```

Пример исписа (**зависи од редоследа извршавања**)

```
[0]{7, 69, 56, 48}
[1]{14, 41, 95, 69}
[2]{77, 37, 55, 65}
[3]{69, 20, 39, 36}
[4]{35, 28, 58, 9}
[5]{25, 91, 7, 1}
[6]{4, 99, 39, 50}
[7]{33, 74, 46, 34}
[8]{73, 37, 52, 51}
[9]{27, 56, 6, 82}
Pocinje izvlacenje:
89 54 44 53 71 64 55 31 23 69 73 42 87 36 76 72 51 6 83 38 43 50 33 97 63 60 45 16 74 24 90 92 94 26 2
8 10 20 17 70 98 65 30 66 86 41 81 47 99 35 58 80 1 39
Dobitni listic: [3]{69, 20, 39, 36}
```

```
public class Skup {
   private class Elem{
      int broj; Elem sled;
      Elem(int i){
         broj = i;
         if(prvi==null) prvi = this;
         else posl.sled = this;
         posl = this;}
   }
   private Elem prvi, posl;
   public boolean dodaj(int i){
      if(postoji(i)) return false;
      new Elem(i);return true;
   }
   public boolean postoji(int i){
      Elem pom = prvi;
      while(pom!=null)
         if(pom.broj == i) return true;
         else pom = pom.sled;
      return false;
   }
   public void prazni(){prvi = posl = null;}
   public String toString(){
      if(prvi == null) return "";
      StringBuilder sb =
        new StringBuilder(prvi.broj);
      Elem pom = prvi;
      while(pom!=null) {
         sb.append(",").append(pom.broj);
         pom = pom.sled;
      }return sb.toString();
   }
}
```

```
public class Generator {
   private int min, max, br;
   private Skup s = new Skup();

   public Generator(int min, int max){
      this.max = max;this.min = min;
   }
   public int sledeci()throws GreskaGenerisanje{
      if(br==max-min+1)
throw new GreskaGenerisanje();
      Random r = new Random();
      while(true){
         int b = r.nextInt(max-min+1) + min;
         if(s.postoji(b)) continue;
         s.dodaj(b);br++;return b;}
   }
   public void obrisiIstorju(){
      br = 0; s.prazni(); }
}
```

```
public class GreskaGenerisanje extends Exception
{
   public String toString(){
      return "Svi brojevi su vec generisani!";}
}
```

```
public class Bubanj extends Thread {
   private Generator g;
   private boolean spremanBroj = false;
   private int br;

   public Bubanj(int min, int max){
      g = new Generator(min, max);
   }
public void pokreni() {start();}
   public void zaustavi() {interrupt();}
   public void run(){
   try{
      while(!interrupted()){
         synchronized (this) {
               while(spremanBroj) wait();
               sleep((int)(Math.random()*100));
               br = g.sledeci();
               spremanBroj = true;
               notify();
            }
         }
      }catch(InterruptedException |
GreskaGenerisanje e){}
   }
   public synchronized int sledeci() {
      while(!spremanBroj)
         try {wait();
         } catch (InterruptedException e) {}
      int ret = br;spremanBroj = false;
      notify();return ret;
   }
   public synchronized boolean
generisan(){return spremanBroj;}
}
```

```
public class Listic {
   private static int sledID = 0;
   private int id = sledID++;
   private int brojevi[], n;
   public Listic(int n){
     brojevi = new int[n];}
   public void dodaj(int i){
     brojevi[n++] = i;}
   public String toString(){
      StringBuilder sb =
        new StringBuilder("["+id+"]");
      for(int i=0;i<n;i++) {
         if(i>0) sb.append(", ");
         sb.append(brojevi[i]);
      }return sb.toString();
   }
   public int uSkupu(Skup s){
      int i = 0;
      for(int br:brojevi) if(s.postoji(br)) i++;
      return i;
   }}
```

```
public class Bingo extends Thread {
   private Bubanj bubanj;
   private Generator gen;
   private Listic listici[];
   private int n, brojeva;
   private Skup s = new Skup();
   private boolean uToku = false;

   public Bingo(int nn, int min, int max, int
brojeva){
      bubanj = new Bubanj(min, max);
      gen = new Generator(min, max);
      listici = new Listic[nn];
      this.brojeva = brojeva;
   }
   public synchronized Listic novi() throws
         GreskaGenerisanje{
      if(uToku) throw new GreskaGenerisanje();
      Listic l = new Listic(brojeva);
      for(int i=0;i<brojeva;i++)
         l.dodaj(gen.sledeci());
      gen.obrisiIstorju();
      return l;
   }
   public synchronized void dodajListic(Listic l)
        throws GreskaGenerisanje {
      if(uToku) throw new GreskaGenerisanje();
      listici[n++] = l;
   }
   public synchronized void pokreni() {
      uToku = true; start();}
   private int max(){
      int m = 0;
      for(Listic l:listici){
         if(l == null) continue;
         int lm = l.uSkupu(s);
         if(lm>m) m = lm;
      }return m;
   }
   public Listic dobitni(){
      for(Listic l:listici)
if(l.uSkupu(s)==brojeva) return l;
      return null;
   }
   public void run(){
      bubanj.pokreni();
      System.out.println("Pocinje izvlacenje:");
      while(!interrupted()){
         int br = bubanj.sledeci();
         System.out.print(br+" ");
         s.dodaj(br);
         int m = max();
         if(m == brojeva){
            uToku = false;
            bubanj.zaustavi();
            break;}}
      if(uToku) {
         bubanj.zaustavi();uToku = false;}
   }}
```