

Izpit Oaps 1 17.1.2008 (Lahko da je kje kaksna napakca, sam tko priblizno je blo!!)

1.naloga

Z metodo binaryInsertion želimo urejati tabele objektov tipa CasovniRok (ni porazred razreda Element), ki predstavlja rok za izvedbo opravila in je določen z atributom datum tipa Datum in Cas tipa Cas. Dopolnite (brez popravljanja) razrede Datum, Cas in CasovniRok ter popravita metodo binaryInsertion tako, da bodo roki po njeni uporabi urejeni kronološko od tistega, ki zapade najprej, do tistega ki zapade najkasneje!!

```
public class Datum //samo dopolnjevanje!!!!
{
private byte dan,mesec; //recimo 19,1
private short leto; //recimo 2008

public Datum(byte dan, byte mesec, short leto) {
this.dan=dan; this.mesec = mesec; this.leto = leto;
}
}
```

```
public class Cas //samo dopolnjevanje !!!
{
private byte ura; //recimo 19
private byte minuta; //recimo 42
```

```
public Cas(byte ura, byte minuta) {
this.ura = ura; this.minuta = minuta;
}
}
```

```
public class CasovniRok //samo dopolnjevanje
{
private Datum datum; //recimo datum(19,1,2008)
private Cas cas; //recimo cas (19.42)
```

```
public CasovniRok(Datum datum, Cas cas) {
this.datum = datum; this.cas = cas;
}
}
public static void binaryinsertion(Element [] a)
{
int i,j,l,r,m;
Element x;
for (i=1; i<a.length; ++i)
{
x=a[i];
l=0;
r=i-1;
while (l<=r)
{
m=(l+r)/2;
if (x.manjsi(a[m]))
r=m-1;
else
l=m+1;
}
for (j=i-1; j>=1; --j)
a[j+1]=a[j];
a[l]=x;
}
}
```

2.naloga

prikazite postopek sortiranja za spodnjo različico optimirane metode quicksorta, ki ureja podano tabelo String s. Pri prikazu izpisite tabelo s na mestih, kjer je metoda (izpisiTabelo(s)), pri tem naj bo razvidno tudi zaporedje rekurzivnih klicev (z levo in desno mejo pozicija)ter izbrani srednji element x .

Sprozitev urejanja:

```
String [] s ={// tabela stevil.....}
qsort(s, 0,11);

public static void qsort(String [] a, int l, int r)
{
String w;

if(r == l+1) {
if (a[l+1].compareTo(a[l])>0){
x=a[l];a[l]=a[l+1];a[l+1]=w;
}
izpisiTabelo(a);
}
else {
int i=l; j=r;
String x=a[l+1];
if(x.compareTo(a[l])>0){
if(x.compareTo(a[l+2])>0)
x=(a[l].compareTo(a[l+2]))>0 ? a[l] : a[l+2]];
}
else
if(a[l+2].compareTo(x)>0)
x = (a[l+2].compareTo(a[l])>0? a[l]: a[l+2]);
do
{
while (a[i].compareTo(x)>0)
++i;
while (x.compareTo(a[j])>0)
--j;
if (i<=j)
{
w=a[i]; a[i]=a[j]; a[j]=w;
++i; --j;
}
} while (i<=j);

izpisiTabelo(a);

if (l<j)
sort(a,l,j);
if (i<r)
sort(a,i,r);
}
}
```

3.naloga

V trgovini hranijo podatke o zneskih plačil prodanega blaga na določen dan v seznamu vrst, ki vsebuje objekte tipa Placilo(tip List<QueueG<Placilo>>). V seznamu je toliko vrst kolikor je blagajn v trgovini! (1 blagajna = 1 vrsta), v vrsti pa so kronološko od zjutraj do večera shranjena vsa plačila določenega dne na tej blagajni!! Predpostavimo, da je trgovina odprta od 8.00 do 20.59 zvečer.

```

public class Placilo{
public int zapStProdaje; //recimo 345
public int ura, minuta;
public double znesek;
//naceloma so to vse spremenljivke ki jih potrebujemo, lahko pa se kaj
dopolnis!!!
}

```

Napisi metodo `public static maxPrometNaUro(List<QueueG<Placilo>> a)` , ki izpise vsoto placil vseh blagajn (recimo 1000 Evrov) in enourno časovno obdobje (recimo 11, za časovno obdobje med 11.00 in 11.59 ure), ko je bila vsota zneskov plačil vseh blagajn največja!! po izvedbi metode mora ostati seznam vrsta nespremenjena. Če rešiš nalogo brez hrambe podatkov v pomožnih tabelah(seznamih, vektorjih), dobis vec tock!!

4.naloga

Podani razred Vreca spremenite tako, da bo generičen in da bo za hranjenje podatkov, namesto tebele objektov, uporabljal vektor desetih vektorjev (`Vector<Vector<T>>`), ki predstavljajo odprto razpršeno tabelo (podobno kot pri tretji seminarski nalogi). Indeks vektorja, v katerem se hrani določen objekt, naj določa vrednost `hashCode()%10`.

```

import java.util.Iterator;

public class Vreca
{
private Object[] vsebina;
private int vel =0;

private void povecaj() {
Object[] temp =vsebina;
vsebina = new Object[temp.length+5];
for(int i=0;i<vel;i++)
vsebina[i]=temp[i];
}

public Vreca () {
vsebina = new Object[5];
}

public void dodaj(Object o) {
if (vel == vsebina.length)
povecaj();
vsebina[vel++] = o;
}

public boolean vsebuje(Object o) {
for(int i=0;i<vel;i++)
if(o.equals(vsebina[i]))
return true;
return false;
}

public void izloci(Object o) {
for(int i =0;i<vel;i++)
if(o.equals(vsebina[i])) {
vsebina[i] = vsebina[--vel];
vsebina[vel]=null;
}
}

public Iterator iterator() {
return new Iterator() {
private int cursor =0;

```

```

public boolean hasNext() {
return cursor<vel;
}

public Object next() {
if(cursor >= vel)
return null;
return vsebina[cursor++];
}
public void remove () {}
};
} //end iterator
} //end class

```

5.naloga

Imel si podan

```
Map<String, String > slovar = new TreeMap<String, String>();
```

Opisi poteka gradnje BST drevesa:

Elemente dodajamo:

```

slovar.put("ena", "one");
slovar.put("dva", "two");
slovar.put("tri", "three");
....
....
...
slovar.put("dvanajst", "twelve");

```

Drevo nato: izpisi : levelorder in inorder;

Potem pa še odstrani elemnete enega za drugim: remove ?
(enajst, tri, pet, dve, ena)!!!!