

Distribuirane mreže i sistemi – Drugi domaći zadatak 2024/25.

Data je implementirana klase Sladiste, kao i aktivna klasa Proizvodjac.

Skladiste.java

```
package proizvodjacpotrosac;
```

```
public class Skladiste{
    private static int statId = 0;
    private int id=++statId;
    private int [] niz;

    private int ulaz;
    private int izlaz;

    private int stanje;
    private final int kapacitet;

    public int getStanje() {
        return stanje;
    }

    public int getId() {
        return id;
    }

    public Skladiste( int kapacitet ) {

        this.kapacitet = kapacitet;

        niz = new int[kapacitet];
    }

    public synchronized void Stavi (int element) throws
    InterruptedException{
        while(stanje == kapacitet) wait();
        niz[ulaz++] = element;
        stanje++;
        if(ulaz == kapacitet)ulaz = 0;
        notifyAll();

    }

    public synchronized int Uzmi() throws InterruptedException{
```

```

        while(stanje==0)wait();
        int element = niz[izlaz];
        niz[izlaz++]=0;
        stanje--;
        if(izlaz == kapacitet) izlaz=0;
        notifyAll();

        return element;
    }
}

```

Proizvodjac.java

```

package proizvodjacpotrosac;

import java.util.logging.Level;
import java.util.logging.Logger;

public class Proizvodjac extends Thread {
    private static int statId = 0;
    private int id=++statId;

    private Skladiste skladiste;
    private int brojac = 0;
    private int minTime ;
    private int maxTime ;

    private int trajanje = minTime + (int)Math.random()*(maxTime-
minTime);

    public Proizvodjac(Skladiste skladiste, int minTime, int maxTime)
    {
        this.skladiste = skladiste;
        this.minTime = minTime;
        this.maxTime = maxTime;
    }

    public void run(){
        System.out.println("Proizvodjac    "+id+"    je    krenuo    sa
proizvodnjom");
        try{
            while(!interrupted()){

                int trajanje = minTime + (int)Math.random()*(maxTime-
minTime);
                sleep(trajanje);
                int proizvod = id*1000 + brojac++;

                skladiste.Stavi(proizvod);
            }
        }
    }
}

```

```

        System.out.println("Proizveden je proizvod" + proizvod);
    }
    } catch (InterruptedException ex) {
        System.out.println("Proizivodjac "+id+ " je završio sa
radom");
    }

}

}

```

Potrebno je na programskom jeziku Java po ugledu na klasu Proizvodjac napraviti klasu Potrosac koji uzima proizvode i skladištai troši ih vremenski period izmedju minTime i maxTime. Takodje, potrebno je napraviti klasu Izvestac koja ima jedinstven automatski generisan celobrojni identifikator i u regularnim vremenskim intervalima prikazuje sadržaj zadanog skladišta. Na kraju, potrebno je napisati glavni program koji kreira i simulira Skladiste od maksimalno 10 proizvoda u koje proizvodi i smešta proizvode 20 proizvođača, uzima i troši 30 potrošača, i postoji tri izveštača koji po rasporedu koji im se dostavi proveravaju sadržaj skladišta i na svakih 10 sekundi ispisuju sadržaj skladišta. Raspored pročitati iz tekstualnog file-a u formatu:

```

1 Izvestac1
2 Izvestac1
3 Izvestac2
4 Izvestac3
5 Izvestac1
.....

100 Izvestac2

```

Simulaciju je moguće prekinuti u bilo kom trenutku.