

Први колоквијум из Објектно оријентисаног програмирања II

- 1) (30 поена) Одговорити концизно (по једна или две реченице) и прецизно на следећа питања:
- Шта су иницијализациони блокови, које врсте постоје и када се извршавају?
 - Да ли имена типова у два потпакета истог пакета могу бити једнака и зашто?
 - У којим контекстима се користи кључна реч `super` и шта означава у сваком од случајева?
- 2) (укупно 70 поена) Саставити на језику *Java* следећи пакет класа:
- (20 поена) **Позориште** има једнозначан аутоматски генерисан целобројан идентификатор и назив који могу да се дохвате. Може да се састави текстуални опис у облику `назив[идент]`.
 - Запослени** у позоришту је описан именом и позориштем у којем је запослен. Сви подаци могу да се дохвате. Може да се дохвати једнословна ознака врсте посла коју обавља, да се састави текстуални опис у облику `име[ознака_посла,назив_позоришта]`.
 - Глумац**, **редитељ** и **костимограф** су запослени у позоришту (ознака врсте посла је **G**, **R** и **K**, респективно).
 - (20 поена) **Представа** је описана називом, позориштем у којем се одржава и садржи произвољан број запослених који учествују у реализацији представе. У реализацији представе могу учествовати само запослени у позоришту у коме се представа одржава, при чему тачно један редитељ, највише два костимографа и произвољан број глумаца. Представа се ствара са редитељем, након чега се костимографи и глумци додају појединачно. Повратна вредност при додавању запосленог је индикатор успеха. Може да се дохвати назив представе, да се дохвати позориште у којем се одржава представа и да се састави текстуални опис представе наводећи назив представе и назив позоришта у којем се представа одржава раздвојене зарезом, а потом запослене који учествују у реализацији представе (сваког запосленог у засебном реду).
 - (20 поена) Позоришни **репертоар** садржи позориште којем припада и може садржати произвољан број представа. Може да се дода представа на репертоар, да се уклони представа задатог назива са репертоара и да се састави текстуални опис репертоара (свака представа у засебном реду).
- (10 поена) Написати на језику *Java* **програм** који направи једно позориште и репертоар за њега, на позоришни репертоар дода представу са једним костимографом и једним глумцем и представу са једним костимографом и три глумца, и испише позоришни репертоар. Користити константне параметре (не треба ништа учитавати).

НАПОМЕНЕ: а) Колоквијум траје **100** минута.

- б) Рад се предаје искључиво у вежбанци за испите (-5 поена за неадекватну вежбанку). Није дозвољено имати поред себе друге листове папира, нити уз себе имати мобилни телефон, без обзира да ли је укључен или искључен.
- в) Водити рачуна о уредности. Нечитки делови текста ће бити третирани као непостојећи. Решења задатака навести по горњем редоследу (-1 поен за лош редослед). Препоручује се рад обичном графитном оловком.
- г) Резултати колоквијума биће објављени на *Web*-у на адреси: `home.etf.rs/~kraus/` (одреднице: *настава* | <име предмета> | *оцене* | *колоквијуми*).

```
// Pozoriste.java
package pozoriste;
public class Pozoriste {
    private static int pozId = 0;
    private int id = ++pozId;
    private String naziv;

    public Pozoriste(String naz) {naziv = naz;}
    public int id() { return id; }
    public String naziv() { return naziv; }
    public String toString()
        { return naziv + "[" + id + "]"; };
}

```

```
// Zaposleni.java
package pozoriste;
public abstract class Zaposleni {
    private String ime; private Pozoriste poz;

    public Zaposleni(String i, Pozoriste p)
        { ime = i; poz = p; }
    public String ime() { return ime; }
    public Pozoriste pozoriste() { return poz; }
    public abstract char vrsta();
    public String toString() { return ime + "["
        + vrsta()+ "," + poz.naziv() + "]"; }
}

```

```
// Glumac.java
package pozoriste;
public class Glumac extends Zaposleni {
    public Glumac(String i, Pozoriste poz)
        { super(i, poz); }
    public char vrsta(){return 'G';}
}

```

```
// Reditelj.java
package pozoriste;
public class Reditelj extends Zaposleni {
    public Reditelj(String i, Pozoriste poz)
        { super (i, poz); }
    public char vrsta(){return 'R';}
}

```

```
// Kostimograf.java
package pozoriste;
public class Kostimograf extends Zaposleni {
    public Kostimograf(String i, Pozoriste poz)
        { super (i, poz); }
    public char vrsta(){return 'K';}
}

```

```
// Predstava.java
package pozoriste;
public class Predstava {
    private static class Elem {
        Zaposleni zaposleni; Elem sled = null;
        Elem(Zaposleni z) { zaposleni = z; }
    }

    private Elem prvi = null, posl = null;
    private String naziv;
    private Pozoriste poz;
    private int brKost = 0;

    public Predstava(String naz, Pozoriste p,
        Reditelj red){
        if (red.pozoriste() != p) System.exit(1);
        naziv = naz; poz = p;
        prvi = posl = new Elem(red);
    }

    public boolean dodaj(Zaposleni z) {
        if (z.pozoriste()!=poz ||
            z.vrsta()=='R' ||
            z.vrsta()=='K' && brKost==2)
            return false;
        posl = posl.sled = new Elem(z);
        if (z.vrsta() == 'K') brKost++;
        return true;
    }

    public String naziv() { return naziv; }
    public Pozoriste pozoriste() { return poz; }
}

```

```
public String toString() {
    String s = "";
    s += naziv + "," + poz.naziv() + "\n";
    for (Elem tek=prvi; tek!=null;
        tek=tek.sled)
        s += tek.zaposleni + "\n";
    return s;
}

```

```
// Repertoar.java
package pozoriste;
public class Repertoar {
    private static class Elem {
        Predstava predstava; Elem sled = null;
        Elem(Predstava p) { predstava = p; }
    }

    private Elem prvi = null, posl = null;
    private Pozoriste poz;

    public Repertoar(Pozoriste p) { poz = p; }

    public boolean dodaj(Predstava p) {
        if (p.pozoriste() != poz) return false;
        Elem novi = new Elem(p);
        if (prvi == null) prvi = novi;
        else posl.sled = novi;
        posl = novi;
        return true;
    }
}

```

```
public boolean ukloni(String naziv) {
    Elem tek = prvi, pret = null;
    while (tek!=null &&
        !tek.predstava.naziv().equals(naziv))
        { pret = tek; tek = tek.sled; }
    if (tek == null) return false;
    if (pret == null) prvi = tek.sled;
    else pret.sled = tek.sled;
    if (tek == posl) posl = pret;
    return true;
}

```

```
public String toString() {
    String s = "";
    for (Elem tek=prvi; tek!=null;
        tek=tek.sled)
        s += tek.predstava + "\n";
    return s;
}

```

```
// Program.java
import pozoriste.*;
public class Program {
    public static void main(String[] argv) {
        Pozoriste p =
            new Pozoriste("Dusko Radovic");
        Repertoar r = new Repertoar (p);
        Predstava p1 = new Predstava(
            "Princeza na zrnu graska", p,
            new Reditelj("Iva Milosevic", p));
        p1.dodaj(new Kostimograf
            ("Maja Mirkovic", p));
        p1.dodaj(new Glumac("Jovo Maksic", p));
        Predstava p2 = new Predstava
            ("Cudne ljubavi", p,
            new Reditelj("Djurdja Tesic", p));
        p2.dodaj(new Kostimograf
            ("Zorana Petrov", p));
        p2.dodaj(new Glumac("Milos Samolov", p));
        p2.dodaj(new Glumac("Jelena Ilic", p));
        p2.dodaj(new Glumac("Sandra Rodic", p));
        r.dodaj(p1); r.dodaj(p2);
        System.out.print(r);
    }
}

```

```
Princeza na zrnu graska,Dusko Radovic
Iva Milosevic[R,Dusko Radovic]
Maja Mirkovic[K,Dusko Radovic]
Jovo Maksic[G,Dusko Radovic]

Cudne ljubavi,Dusko Radovic
Djurdja Tesic[R,Dusko Radovic]
Zorana Petrov[K,Dusko Radovic]
Milos Samolov[G,Dusko Radovic]
Jelena Ilic[G,Dusko Radovic]
Sandra Rodic[G,Dusko Radovic]
```