Zadaci za pripremu kolokvijuma

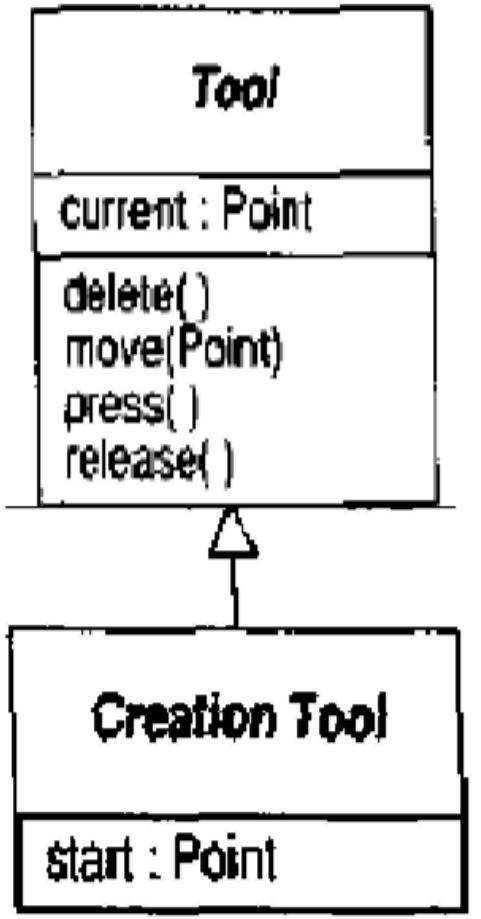
Zadaci su prezeti iz knjige

Practical Object-Oriented Design with UML, ***Mark Priestly***

Zadatak 1

- Skicirati implementacije dijagrama

a)



b)

c)

Zadatak 1, rješenje

a)

```
public  abstract class Tool {
    Point current;

    public abstract void move(  Point p)   ;
    public  abstract   class CreationTool  extends Tool {
    Point  start   ;
    public void move( Point p ) { ... }
}
```

b)

```
public class Diagram
{
    private Vector elements = new Vector(16};
    public void add( Element e )
    {
        elements.addElement(e);
    }
}
```

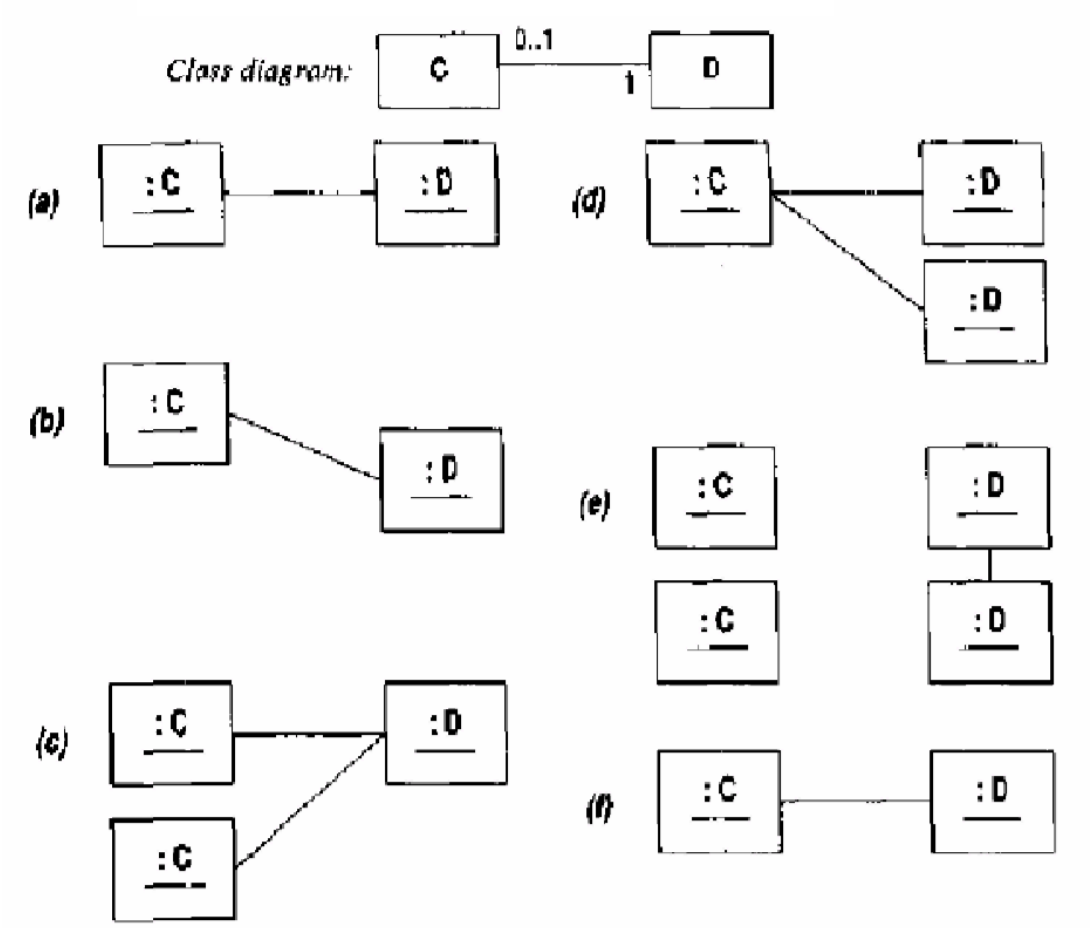
c)

```
public abstract class {
    Diagram diagram ;

    Tool (Diagram d) {
        if (d == null) {
            // throw NullDiagramError
        }
        diagram = d;
```

Zadatak 2

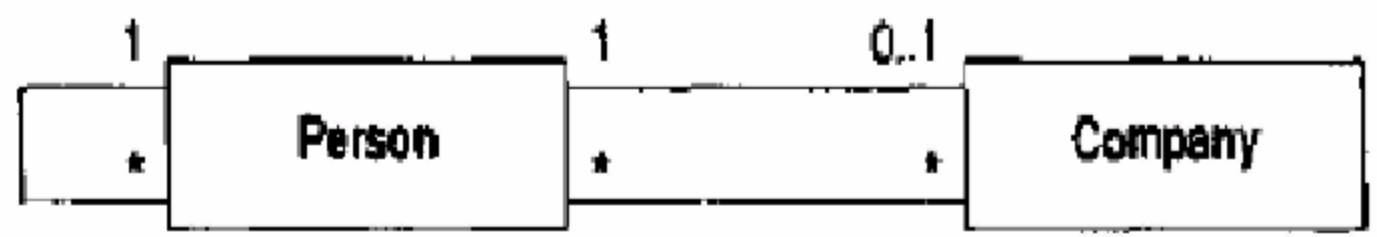
Za dati dijagram klasa, navesti ispravne objektne dijagrame



Zadatak 3

- Nacrtati dijagram klasa i skicirati implementaciju u jeziku C++ prema sljedećem opisu.
- Kompanija može da ima više zaposlenih. Svaka kompanija ima izvršnog direktora, a svaki zaposleni direktnog nadređenog. Nadređeni može da bude šef većem broju radnika.

Zadatak 3, uputstvo za rješavanje



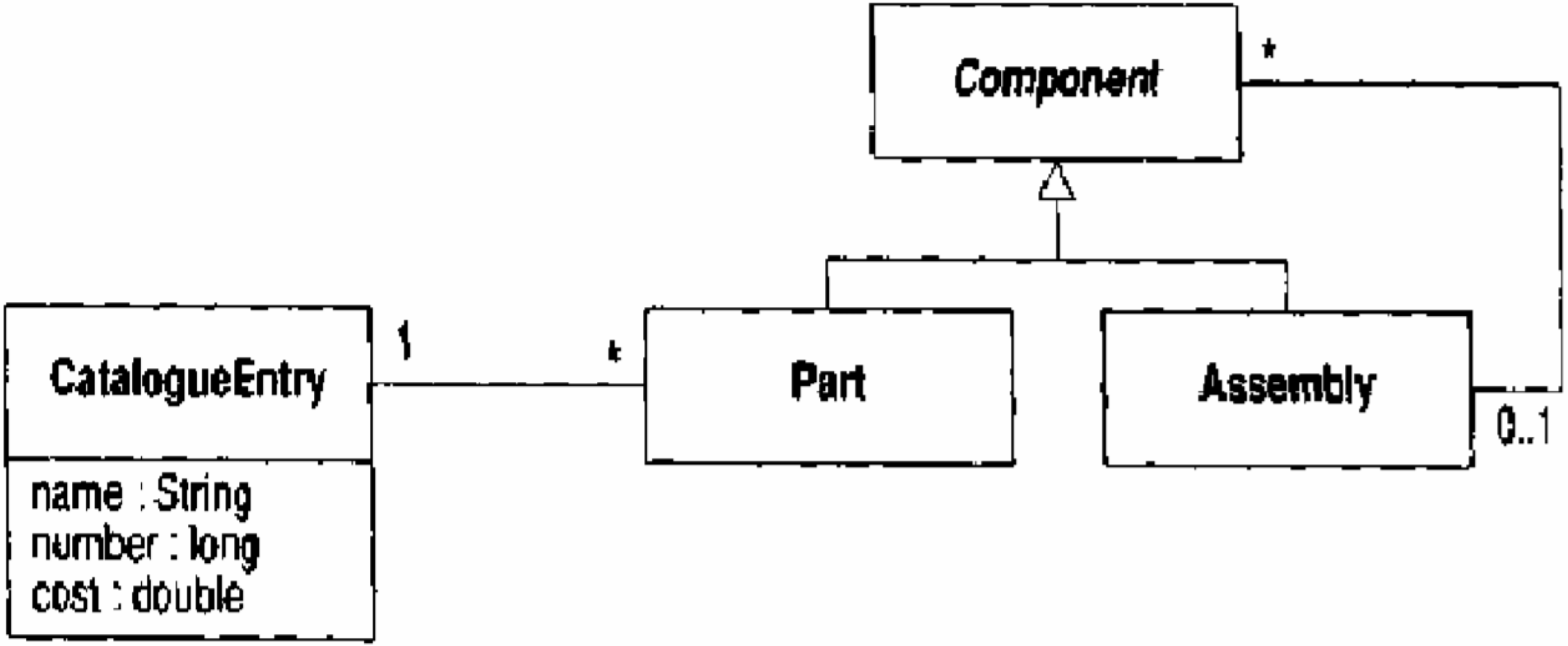
Zadatak 4

Dizajnira se pojednostavljeni softverski modul za upravljanje magacinom. Za svaki dio koji se čuva u magacinu potrebno je znati tip, naziv i cijenu. Djelovi mogu da se kombinuju i na taj način kreiraju složene djelove. Složeni dio može da se sastoji iz proizvoljnog broja djelova, koji mogu takođe biti složeni (hijerarhijsko uređenje). Sistem treba da podrži računanje cijene za djelove i da prikaže spisak pripadajućih djelova.

Nacrati dijagram klasa.

Skicirati implementaciju u jeziku C++.

Zadatak 4, rješenje (1)



Zadatak 4, rješenje (2)

- `CatalogEntry` je klasa koja predstavlja tip

```
public  class CatalogueEntry
{
    public CatalogueEntry (String nm, long nun, double c
    {
        name = nm;
        number = num;
        cost = cst;
    }
        public string getName(){return name;}
        public long getNumber() { return number; }
        public double getCost() { return cost; }

        private string name;
        private long number;
        private double cost;
}
public class Part
{
    public Part (CatalogueEntry  e) {
        entry = e ;
    }
    private CatalogueEntry entry  ;
}
```

Zadatak 4, rješenje (3)

- U klasi `Assembly` treba napisati i metodu `addComponent`

```
public abstract class Component
{
    public abstract double cost ();
}
public class Part extends Component
{
    public double cost()
    {
        return entry,getCost();
    }
}
public class Assembly extends Component
{
    private Vector components = new Vector() ;
    public double cost() ]
    {
        double total = 0.0 ;
        Enumeration enum = components.elements();
        while (emum.hasMoreElements())
        {
            total += ((Component) enum.nextElement()).cost();
        }
        return total ;
    }
}
```

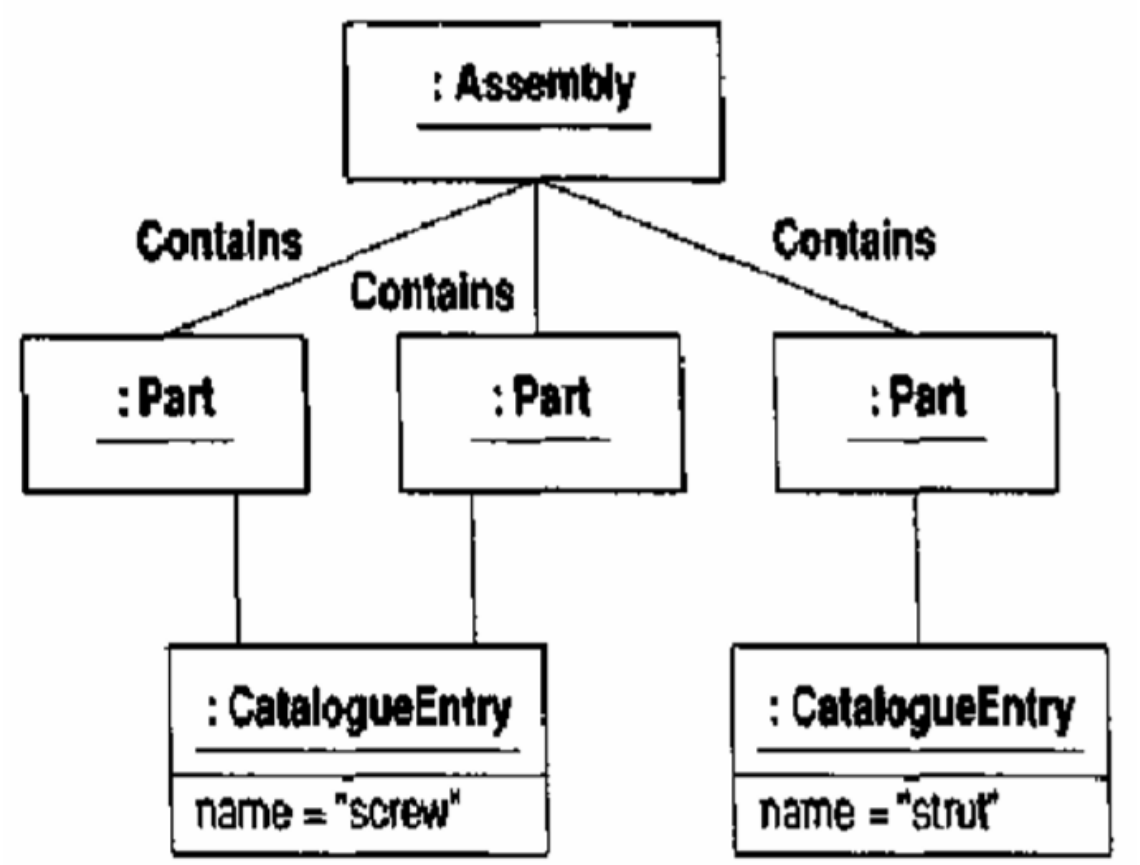
Zadatak 5

- Nacrati objekti dijagram poslije izvršavanja dijela koda 1
- Nacrtati objektni dijagram poslije izvršavanja dijela koda 2
- Na prethodnom dijagramu označiti poruke koje se šalju pozivom metode `a.cost()`

1.
```
CatalogueEntry  frame
= new CatalogueEntry{"Frame",10056, 49.95);
CatalogueEntry screw
= new CatalogueEntry("Screw",28834, 0.02);
CatalogueEntry  spoke
=  new CatalogueEntry("Spoke",47737, 0,95);
Part screwl   = new Part(screw);
Part screw2   = new Part(screw);
Part theSpoke = new Part(spoke);
```

2.
```
Assembly a = new Assembly() ;
a.add(screwl);
a.add(screw2);
a.add(theSpoke);
```

Zadatak 5, uputstvo za rješavanje



Zadatak 6

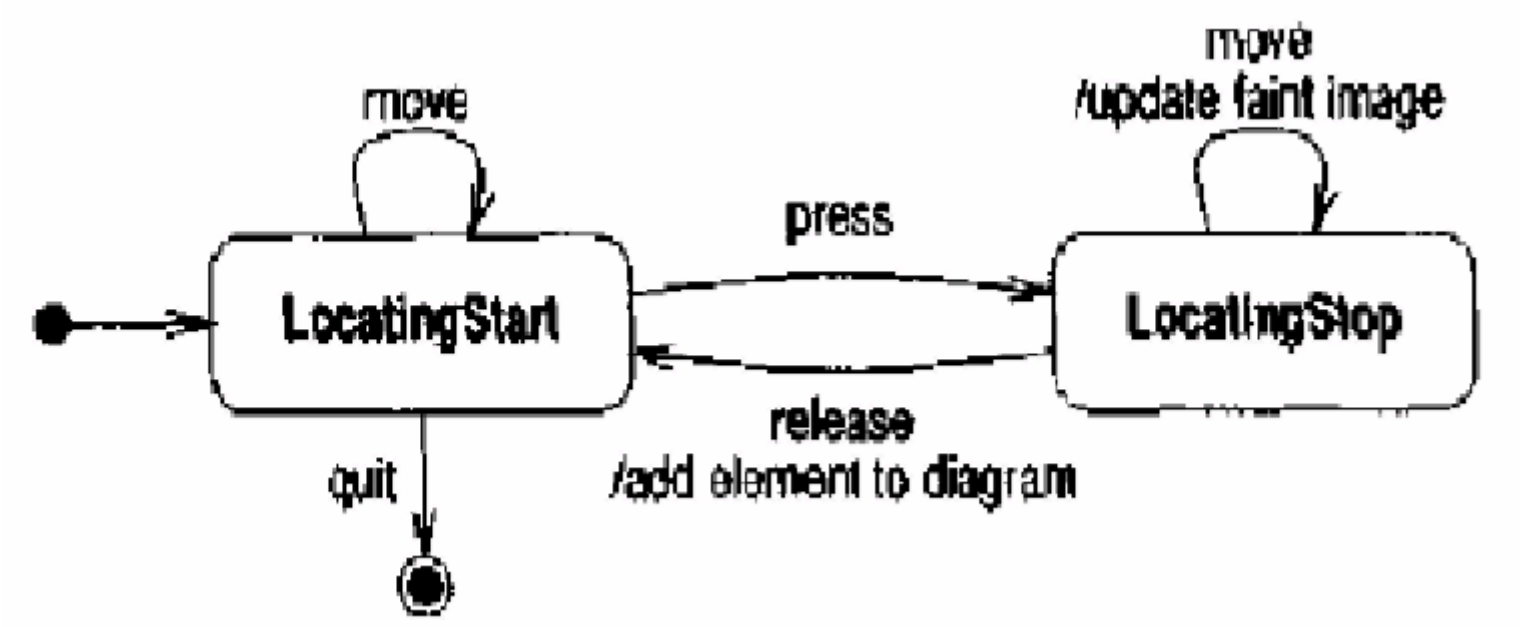
- Na mašini su trenutno prijavljeni korisnici A, B, i C. Pokrenuta su četiri procesa 1001, 1002, 1003 i 1004. Korisnik A je pokrenuo procese 1000 i 1001, korisnik B proces 1003, korisnik C pokrenuo je proces 1004.
  - Nacrtati dijagram klasa
  - Nacrtati objektni dijagram sa linkovima koji će pokazati da se procesi izvršavaju na mašini i koji korisnici su ih pokrenuli
  - Razmatra se operacija koja treba da prikaže spisak svih procesa koji su aktivni na određenoj mašini i za određenog korisnika. Prikazati koje se poruke razmjenjuju između objekata prikazanih na prethodnom dijagramu.

Zadatak 7

- Nacrtati use case dijagram za pojednostavljeni bibliotekarski sistem. U biblioteci se mogu iznajmiti knjige i časopisi. Knjige mogu da iznajmljuju članovi biblioteke. Časopise mogu da iznajme samo zaposleni u biblioteci. Prilikom iznajmljivanja knjige provjerava se da li taj član biblioteke kasni sa vraćanjem neke već iznajmljene knjige ili je već iznajmio dozvoljeni broj knjiga. Članovi biblioteke mogu sami provjeravati stanje na svojoj članskoj karti. Bibliotekar može da provjeri člansku kartu svih članova. Član biblioteke može da rezerviše knjigu koja je već izdata. Kada se napravi četvrta rezervacija jedne knjige, bibliotekar pokreće proceduru za njenu nabavku.

Zadatak 8

- Napisati implementaciju dijagrama stanja za klasu `CreationTool`



Zadatak 8, rješenje (1)

```
public abstract class GreationTool extends Tool (
    final static int LocatingStart = 0 ;
    final static int LocatingStop = 1 ;
    int state ;

    CreationTool(Diagram d) (
        super(d) ;
        state = LocatingStart;
    }
    ...
}
```

Zadatak 8, rješenje (2)

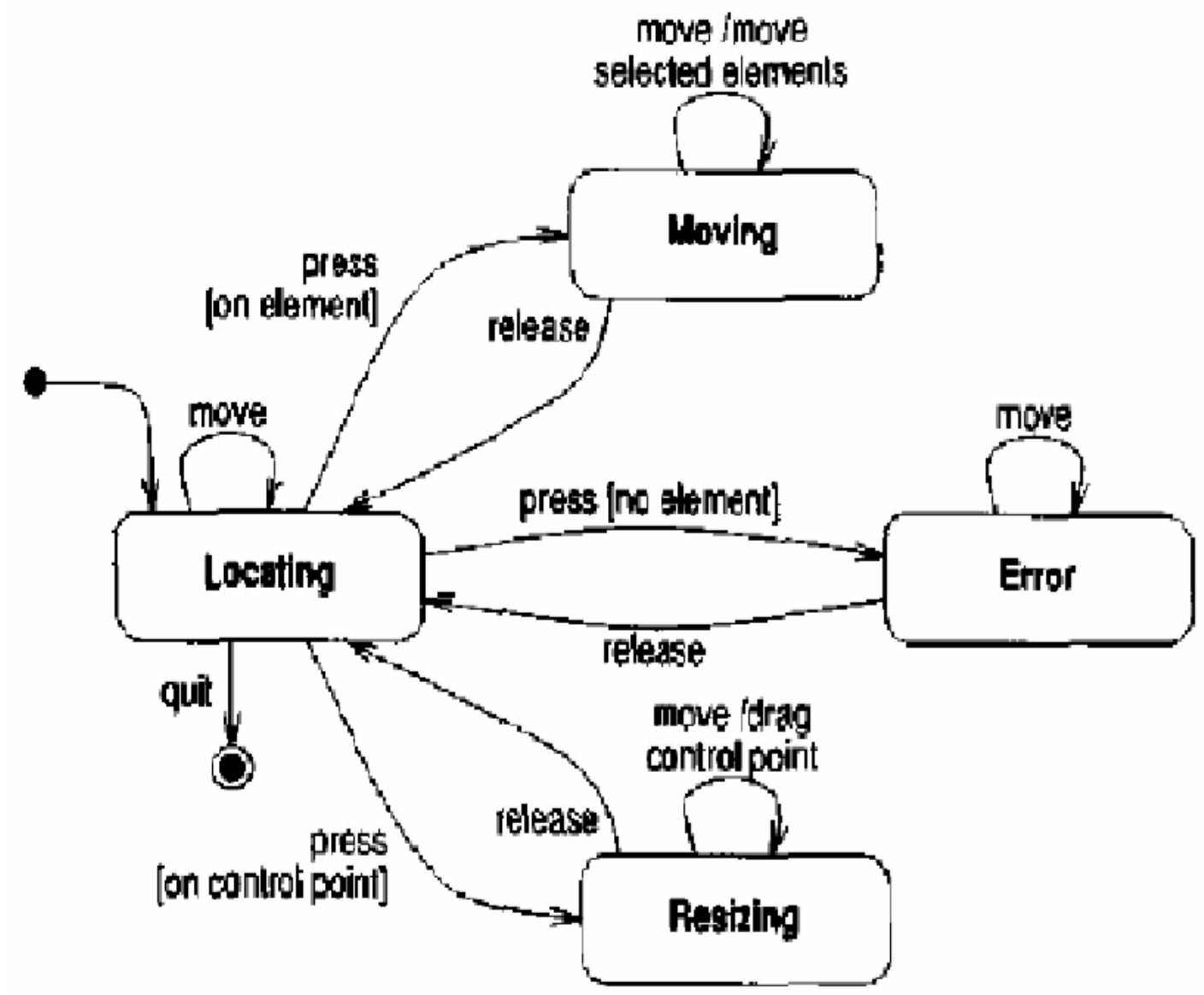
```
public void operation() (
    switch (state) {
        case LocatingStart:
        // Specific actions for locating start state break ;
        case LocatingStop:
        // Specific actions for locating stop state break ;
    }
}
```

Zadatak 8, rješenje (3)

```
public void release() {
switch (state) (
    case LocatingStart :
    break ;
    case LocatingStop:
        Element e = newElement (start, current) ;
        diagram. add(e) ;
        state = LocatingStart ;
        break ;
    }
}
```

Zadatak 9

Napisati implementaciju dijagrama stanja za klasu `SelectionTool`



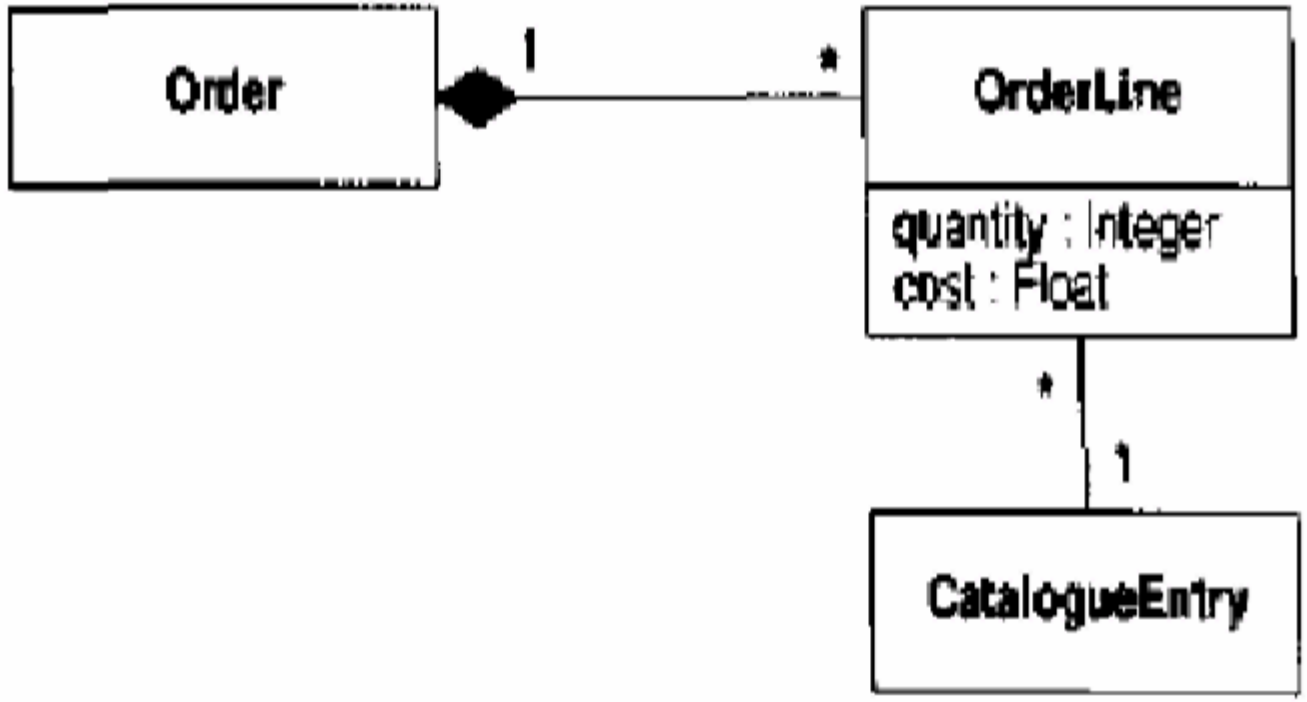
Zadatak 9, rješenje

```
public void press (} {
    switch   (state)    {
        case Locating:
            if   (findcontrol(current))
            {
                state  = Resizing;
            }
            else  if  (diagram.find(current)!= null)
            {
                state  = Moving ;
            }
            else
            {
                state = Error ;
            }
        case Moving:
            break  ;
        case Resizing:
            break ;
        case Error:
            break ;
```

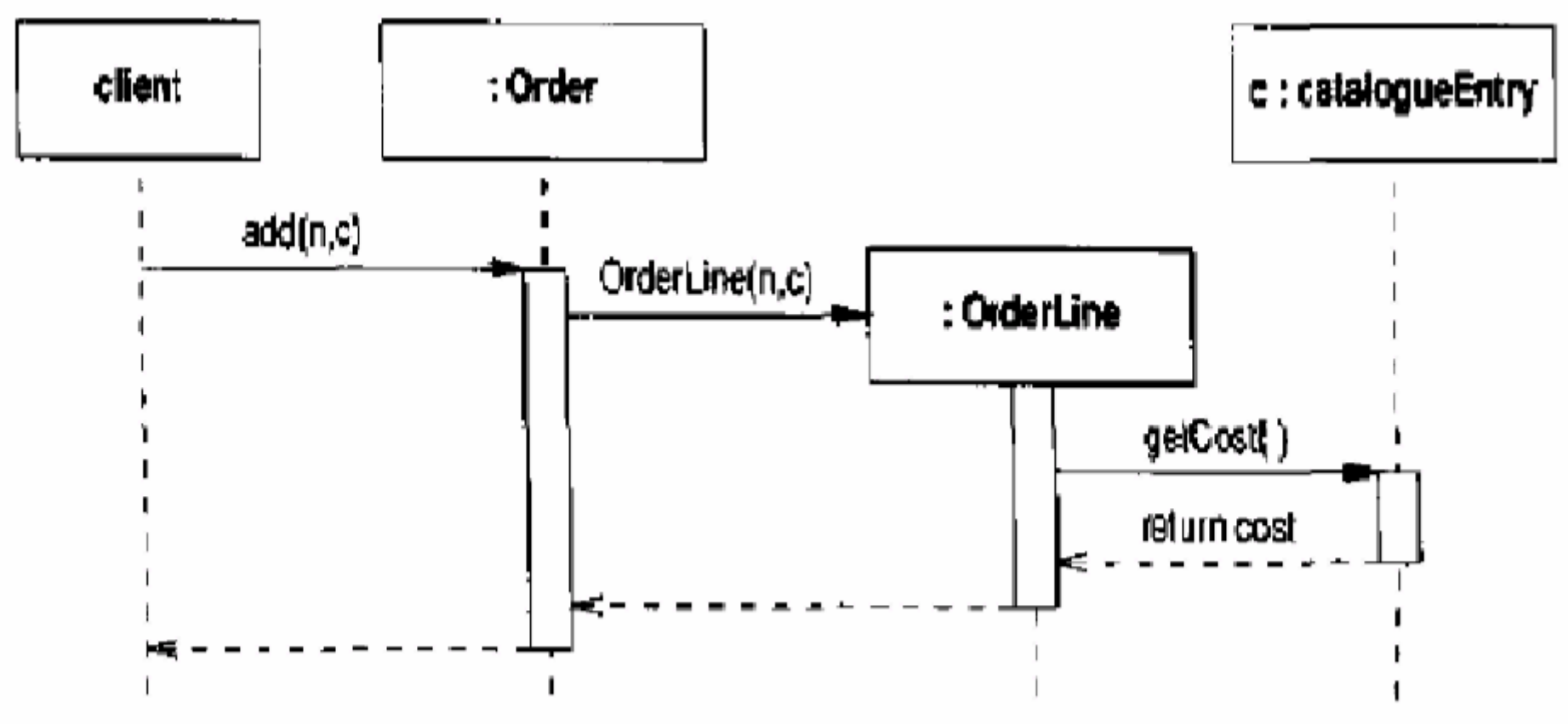
Zadatak 10

- Sistem za upravljanje magacinom proširuje se sa modulom koji obrađuje narudžbenice. Svaka narudžbenica sastoji se od više stavki. Za stavku se zna količina i ukupna cijena proizvoda na koji se odnosi.
  - Nacrtati dijagram klasa i skicirati implementaciju u jeziku C++.
  - Nacrtati dijagram sekvence za metodu `add` klase `Order` kojom se dodaje stavka predstavljena klasom `OrderLine` u narudžbenicu i skicirati implementaciju u jeziku C++.
  - Pretpostaviti da `CatalogEntry` sadrži broj raposloživih djelova u magacinu. Modifikovati prethodni dijagram i implementaciju tako da se prilikom kreiranja nove stavke provjerava da li postoji dovoljno djelova.

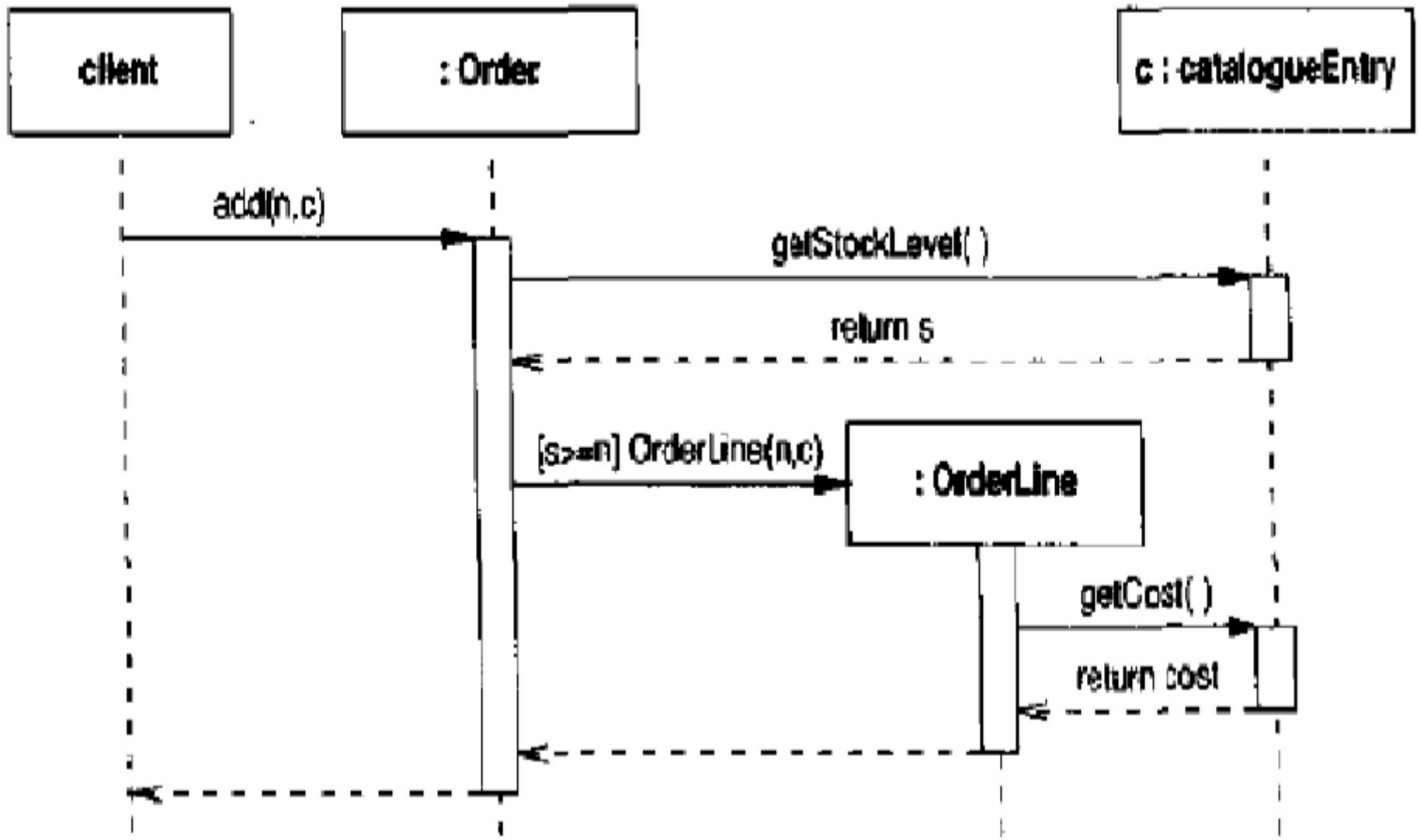
Zadatak 10, uputstvo za rješavanje (1)



Zadatak 10, uputstvo za rješavanje (2)



Zadatak 10, uputstvo za rješavanje (3)



Zadatak 11

- Pokazati da sljedeća dva dijagrama nijesu ekvivalentna. Modifikovati dijagram kako bi se postigla ekvivalentnost.