Data och Informationsteknik / Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
Ansvarig lärare: Magnus Myréen

DAT050 Objektorienterad programmering

Modellsvar för tentamen, måndag, 2020-01-04, 08:30-12:30

Vitsordsgränser: 3=24p, 4=36p, 5=48p, max 60p.

Kom ihåg att inte fastna på en uppgift. Bestäm i förväg din egen tidsgräns per uppgift. ***Lycka till!***

Modellsvar för Uppgift 1: [15 poäng totalt] *att skriva klass, toString*

```
import java.util.*;

public class SchoolClass {

    private String teacher;
    private String className;
    private TreeSet<String> pupils;  // TreeSet gör så att iteratorn ger sorterade namn

    public SchoolClass(String className, String teacher) {
        this.teacher = teacher;
        this.className = className;
        this.pupils = new TreeSet<String>();
    }

    public void addPupil(String name) {
        pupils.add(name);
    }

    public String toString() {
        String ret = "[ SchoolClass " + className + ": " + teacher + " teaches";
        Iterator<String> it = pupils.iterator();
        String sep = " ";
        while (it.hasNext()) {
            ret = ret + sep + it.next();
            sep = ", ";
        }
        return ret + " ]";
    }

}
```

Man kan använda andra datastrukturer än `TreeSet`, men då behöver man se till att returvärdet från `toString` har namnen sorterade i alfabetisk ordning, t.ex. med hjälp av `Collections.sort`.

Modellsvar för Uppgift 2: [10 poäng totalt] *private/public, immutable*

Koden för en klass beror ofta på att vissa relationer håller mellan instansvariablerna i klassen. Om instansvariablerna är deklarerade som något annat än `public` så kan utomstående ändra på dem utan att klassens kod vet om det. Då kan utomstående kod göra så att klassens kod inte fungerar som den ska.

Exempel: Ifall `RatNum` klassen från Lab 2 har sina instansvariabler deklarerade på följande sätt:

```
public class RatNum {

    public int n;
    public int d;

...
```

Då kommer följande kod som försöker kolla om 1/1 och 2/2 är samma tal ge `false`, vilket är fel!

```
RatNum a = new RatNum(1);
RatNum b = new RatNum(1);
b.n = 2;
b.d = 2;
System.out.println(a.equals(b));
```

Problemet uppstår p.g.a. klassens kod antar att gcd mellan `n` och `d` är 1.

Modellsvar för Uppgift 3: [15 poäng totalt] *att skriva klass, interface, exceptions*

Del a)

Följande klass kan användas som ett exception för att den ärver `Exception`.

```
public class NoSuchCarException extends Exception { }
```

Del b)

```
public Car whichCarToBuy(List<Car> cars, int maxMoneyToSpend)
          throws NoSuchCarException {
    if (cars == null) {
        throw new NoSuchCarException();
    }
    Car max = null;
    Iterator<Car> it = cars.iterator();
    while (it.hasNext()) {
        Car c = it.next();
        if (c != null) {
            if (c.getPrice() <= maxMoneyToSpend) {
                if (max == null) {
                    max = c;
                } else if (max.getPrice() < c.getPrice()) {
                    max = c;
                }
            }
        }
    }
    if (max == null) {
        throw new NoSuchCarException();
    } else {
        return max;
    }
}
```

Modellsvar för Uppgift 4: [10 poäng totalt] *testning, int typen, exceptions*

```
public boolean isOK(MyInterface mi) {
    int i = 1;
    while (i != 0) {
        try {
            mi.myFunction(i);
        } catch (Exception e) {
            return false;
        }
        i++;
    }
    return true;
}
```

Modellsvar för Uppgift 5: [10 poäng totalt] *arv, händelsehantering, Javas API*

```
import java.util.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class RandomColorButton extends JButton implements ActionListener {

    private Random r = new Random();

    public RandomColorButton() {
        super();
        addActionListener(this);
        this.setOpaque(true);  // denna rad kanske behövs för testning på Mac OS
    }

    public void actionPerformed(ActionEvent e) {
        Color c = new Color(r.nextInt(256),r.nextInt(256),r.nextInt(256));
        this.setBackground(c);
    }

}
```