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, torsdag, 2022-10-27, 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: [10 poäng] *grunder i objektorientering*

A. Objekt som är immutable går inte att ändra. Istället för att ändra på immutable objekt skapas nya objekt som representerar det ändrade objektet.

B. Regler för skrivning av immutable klasser: man får aldrig ändra på instansvariabler efter att ett objekt har skapats; man får inte ha referensvärden till muterbara objekt som andra kanske har referensvärden till (klona!); som alltid: alla instansvariabler bör vara private.

C. Man undviker aliasing problem. Aliasing problem uppstår när två olika delar av ett program har referenser till ett objekt och den ena delen av programmer ändrar på objektet på ett sätt som den andra delen av programmet inte förväntar sig. Med immutable undviker man aliasing problem för att det inte går att ändra på objekt.

D. (hög) kohesion = att varje klass har ett väldefinierat syfte
(låg) koppling = att klasser inte är onödigt beroende på varandra

E. Model känner inte till (har ingen koppling till) View eller Control
View känner till Model, men ändrar inte i Model; View känner inte till Control
Control känner till Model och View

Modellsvar för Uppgift 2: [9 poäng totalt] *arv, referensvärden*

Följande skrivs ut:

```
test in A
false
abc
```

Modellsvar för Uppgift 3: [16 poäng totalt] *att skriva klasser, testning, exceptions*

A.

```
public int howMany(int j) {
    int count = 0;
    for (Integer i : bag) {
        if (j == i) { count++; }
    }
    return count;
}
```

B.

```
    try {
        IntBag bag = new IntBag();
        bag.insert(1);
        bag.insert(5);
        bag.insert(1);
        bag.insert(1);
        bag.insert(2);
        assertEquals(3,bag.howMany(1));
    } catch (Exception e) {
        assertTrue(false);
    }
```

C.

```
public boolean equals(Object obj) {
    if (obj == null) { return false; }
    if (obj.getClass() != this.getClass()) { return false; }
    IntBag other = (IntBag)obj;
    return this.subset(other) && other.subset(this);
}

private boolean subset(IntBag other) {
    for (Integer i : this.bag) {
        if (other.howMany(i) < this.howMany(i)) { return false; }
    }
    return true;
}
```

Modellsvar för Uppgift 4: [10 poäng] *interface, enum, exceptions*

```
public CanCompare findLeast(CanCompare[] array) throws EmptyException {
    if (array == null) {
        throw new EmptyException();
    }
    int least = -1;
    for (int i=0; i<array.length; i++) {
        if (array[i] != null) {
            if (least < 0) {
                least = i;
            } else {
                CanCompare c = array[i];
                if (c.compare(array[least]) == CompareResult.LESS) {
                    least = i;
                }
            }
        }
    }
    if (least < 0) {
        throw new EmptyException();
    } else {
        return array[least];
    }
}
```

Modellsvar för Uppgift 5: [15 poäng totalt] *grafiskt användargränssnitt, lyssnare*

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

public class RedGreen extends JFrame {

    private String output = "";

    public RedGreen() {
        super("RedGreen");
        JPanel p = new JPanel();
        p.setLayout(new FlowLayout());
        JButton button1 = new JButton("red");
        button1.addActionListener(
           (e) -> { output += "red "; System.out.println(output); });
        JButton button2 = new JButton("green");
        button2.addActionListener(
           (e) -> { output += "green "; System.out.println(output); });
        p.add(button1);
        p.add(button2);
        this.add(p);
        this.setVisible(true);
        this.pack();
    }

    public static void main(String[] args) {
        new RedGreen();
    }

}
```