

DAT050 Objektorienterad programmering

Modellsvär för tentamen, måndag, 2022-01-03, 08:30-12:30

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

Handledaren Jessica Barai besöker tentan och svarar på eventuella frågor om uppgifterna.
Kom ihåg att inte fastna på en uppgift. Bestäm i förväg din egen tidsgräns per uppgift. **Lycka till!**

Modellsvär för Uppgift 1: [10 poäng totalt] *grunder i objektorientering*

- A. Begreppet "black box tänkande" avser att användaren av ett objekt inte bör veta eller ha tillgång till hur ett objekt internt är implementerat.
- B. Välskrivna objektorienterade program kan sägas vara modulära ifall klasserna har *hög kohesion* och *låg koppling*. Detta innebär att delarna av programmet kan lätt bytas ut mot andra delar och programmets delar är återanvändbara i andra program.
- C. Här används double:

```
public class Average {  
  
    private double sum = 0.0;  
    private double count = 0.0;  
  
    public void addInt(int n) {  
        sum = sum + (double)n;  
        count = count + 1;  
    }  
  
    public double getAverage() {  
        return sum / count;  
    }  
}
```

Modellsvar för Uppgift 2: [10 poäng totalt] *gränssnitt*

```
public class SafeSwitch implements Switch {  
    private Switch s;  
  
    public SafeSwitch(Switch s) {  
        this.s = s;  
    }  
  
    public void turnOn() {  
        if (!s.isOn()) {  
            s.turnOn();  
        }  
    }  
  
    public void turnOff() {  
        if (s.isOn()) {  
            s.turnOff();  
        }  
    }  
  
    public boolean isOn() {  
        return s.isOn();  
    }  
}
```

Modellsvar för Uppgift 3: [15 poäng totalt] *listor*

```
public List<Integer> pascalsTriangleLine(int n) {  
    if (n < 1) {  
        throw new IllegalArgumentException("Bad input");  
    }  
    List<Integer> l = new LinkedList<Integer>();  
    l.add(1);  
    for(int k=1; k < n; k++) {  
        List<Integer> tmp = new LinkedList<Integer>();  
        tmp.add(1);  
        for(int i=1; i<l.size(); i++) {  
            tmp.add(l.get(i-1) + l.get(i));  
        }  
        tmp.add(1);  
        l = tmp;  
    }  
    return l;  
}
```

Modellsvar för Uppgift 4: [10 poäng totalt] *string, equals, referensvärden*

A. Programmet skriver ut:

```
false
true
false
false
true
```

B. Det är viktigt att använda equals när man jämför strängar (på sista raden).

```
public boolean equals(Object other) {
    if (other == null) { return false; }
    if (other.getClass() != this.getClass()) { return false; }
    StringPair o = (StringPair)other;
    return this.a.equals(o.a) && this.b.equals(o.b);
}
```

Modellsvar för Uppgift 5: [15 poäng totalt] *model-view-controller, grafik*

```
public class SnakeView extends JPanel {

    private SnakeModel model;

    public SnakeView(SnakeModel model) {
        this.model = model;
    }

    public void paintComponent(Graphics g) {
        g.setColor(Color.BLACK);
        g.fillRect(0, 0, getWidth(), getHeight());
        for (int i=0; i < 20; i++) {
            for (int j=0; j < 20; j++) {
                Cell c = model.getCellAt(i,j);
                if (c == Cell.SNAKE) {
                    g.setColor(Color.WHITE);
                    g.fillRect(i*30, j*30, 30, 30);
                } else if (c == Cell.FOOD) {
                    g.setColor(Color.RED);
                    g.fillOval(i*30, j*30, 30, 30);
                }
            }
        }
    }
}
```