

DAT050 Objektorienterad programmering

Modellsvar för tentamen, onsdag, 2021-08-25, 14:00-18:00

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 totalt]

- A) Nyckelordet `private` skrivs på instans- och klassvariabler, samt metoder. Dessa med detta nyckelord kan inte komma åt utifrån klassen och ej heller från subklasser. Det används för att implementera inkapsling, se del B.
- B) Inkapsling är ett centralt koncept i objektorientering. Det betyder att användaren av en klass inte ska kunna se eller använda sig av det interna i en klass. Det som syns utifrån av en klass är dess gränssnitt. Man kan deklarera gränssnitt med nyckelordet `interface`.
- C) UML diagram är diagram som består av lådor och pilar med annotationer som symboler och text. Idén med UML diagram är att de ger en mera abstrakt vy av programmet, speciellt hur klasser relaterar till varandra.

Modellsvar för Uppgift 2: [totalt 15 poäng]

```
public class LottoRow {

    private int[] row = new int[7];

    public LottoRow(int[] a) {
        if (a == null) {
            throw new IllegalArgumentException("Null");
        }
        if (a.length != 7) {
            throw new IllegalArgumentException("Bad length");
        }
        for (int i=0; i<7; i++) {
            for (int j=1; j<7; j++) {
                if (a[i] == a[j]) {
                    throw new IllegalArgumentException("Duplicate");
                }
            }
            row[i] = a[i]; // copies content to avoid aliasing
        }
    }

    public String toString() {
        String res = "" + row[0];
        for (int j=1; j<7; j++) {
            res = res + ", " + row[j];
        }
        return res;
    }

    public int countMatches(LottoRow other) {
        if (other == null) {
            return 0;
        }
        int r = 0;
        for (int i=0; i<7; i++) {
            for (int j=0; j<7; j++) {
                if (row[i] == row[j]) {
                    r = r+1;
                }
            }
        }
        return r;
    }
}
```

Modellsvar för Uppgift 3: [10 poäng totalt]

```
public List<Integer> addChecksum(List<Integer> l) {
    List<Integer> res = new LinkedList<Integer>();
    Iterator<Integer> it = l.iterator();
    int sum = 0;
    while (it.hasNext()) {
        int k = it.next();
        res.add(k);
        sum = ((k % 7001) + sum) % 7001; // avoids overflow in +
    }
    res.add(sum);
    return res;
}
```

Modellsvar för Uppgift 4: [10 poäng totalt]

```
public static int maxValueOf(Tree t) {
    int max = t.getValue();
    Iterator<Tree> it = t.getSubtrees();
    while (it.hasNext()) {
        Tree sub = it.next();
        int temp = maxValueOf(sub);
        if (temp > max) {
            max = temp;
        }
    }
    return max;
}
```

Modellsvar för Uppgift 5: [15 poäng totalt]

```
public class Click extends JFrame implements MouseListener {

    private long prevClick;

    public Click() {
        JPanel p = new JPanel();
        p.addMouseListener(this);
        prevClick = System.currentTimeMillis();
        add(p);
        pack();
        setVisible(true);
    }

    public void mouseClicked(MouseEvent e) {
        long newTime = System.currentTimeMillis();
        System.out.println("Time: " + (newTime - prevClick) + " ms");
        prevClick = newTime;
    }

    public void mouseEntered(MouseEvent e) {}
    public void mouseExited(MouseEvent e) {}
    public void mousePressed(MouseEvent e) {}
    public void mouseReleased(MouseEvent e) {}

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