

Lösningsförslag till tentamen

P r e l i m i n ä r

Kurs
Tentamensdatum

Objektorienterad programmering
2011-10-20

Program
Läsår
Examinator

D2
2011/2012, lp 1
Uno Holmer

Uppgift 1 (10 p)

Ingen lösning ges. Se kurslitteraturen.

Uppgift 2 (8 p)

```
public class DieButton extends JButton implements ActionListener {
    private DieIcons icons;
    private static Random random = new Random();

    public DieButton() {
        icons = DieIcons.getInstance();
        addActionListener(this);
        roll();
    }
    private void roll() {
        setIcon(icons.getIcon(random.nextInt(icons.size())));
    }
    public void actionPerformed(ActionEvent e) { roll(); }
}
```

Uppgift 3 (10 p)

```
public class CardHand {
    ...

    @Override
    public CardHand clone() {
        CardHand result = null;
        try {
            result = (CardHand)super.clone();
            result.cards = (ArrayList<Card>)cards.clone();
            for ( int i = 0; i < cards.size(); i++ )
                result.cards.set(i, cards.get(i).clone());
            return result;
        }
        catch (CloneNotSupportedException e) {
            throw new InternalError();
        }
    }
}
```

v.g.v.

```
@Override
public final boolean equals(Object o) {
    if ( this == o )
        return true;
    else if ( o instanceof CardHand ){
        CardHand other = (CardHand)o;
        ArrayList<Card> left = (ArrayList<Card>)cards.clone();
        ArrayList<Card> right =
            (ArrayList<Card>)other.cards.clone();
        Collections.sort(left);
        Collections.sort(right);
        return left.equals(right);
    }
    return false;
}
}
```

Uppgift 4 (10 p)

```
public class ListMindMap implements MindMap {
    private String description;
    private ListMindMap parent;
    private List<MindMap> subtopics;

    public ListMindMap(String description) {
        setDescription(description);
        parent = null;
        subtopics = null;
    }
    public Iterator<MindMap> iterator() {
        return subtopics.iterator();
    }
    public MindMap addSubtopic(String description) {
        ListMindMap t = new ListMindMap(description);
        t.setParent(this);
        if ( subtopics == null )
            subtopics = new LinkedList<MindMap>();
        subtopics.add(t);
        return t;
    }
    ...
    public void print() {
        print(0,"1");
    }
    private void print(int level,String index) {
        print(level,index,description);
        if ( subtopics != null ) {
            int i = 1;
            for ( MindMap t : subtopics )
                if ( t instanceof ListMindMap )
                    ((ListMindMap)t).print(level+1,index + "." + i++);
        }
    }
    private void print(int level,String index,String text) {
        for ( int i = 0; i < 3*level; i++ )
            System.out.print(' ');
        System.out.println(index + " " + text);
    }
}
```

Uppgift 5 (10 p)

1. I `func`: Anropet `obj.f4()` är ej korrekt. `obj` har typen `C2` men `f4` är inte känd i vare sig `C1` eller `C2`.
2. I `main`: Anropet `func(new C1())` är fel. `C1` är abstrakt och får ej instansieras.
3. I `main`: Anropet `func2(new C2())` är ej typkorrekt. I `func2` har den formella parametern typen `C3`, men `C2` är ej subtyp till `C3`.

Om vi tar bort ovanstående blir utskriften (1p per rätt svar):

```
C2.f1
C2.f2
C1.f3
++C2.f1++      print överskuggas i C3
++C3.f2++
++C1.f3++
++C3.f4++
```

Uppgift 6 (4+7+1 p)

a) (4 p)

```
public class CellView extends JButton implements Observer {
    public CellView(int size) {
        setPreferredSize(new Dimension(size,size));
        setBackground(Color.LIGHT_GRAY);
    }

    public void update(Observable o, Object arg) {
        if ( o instanceof Cell ) {
            switch ( ((Cell)o).getType() ) {
                case FLOOR:
                    setBackground(Color.WHITE); break;
                case WALL:
                    setBackground(Color.BLACK); break;
            }
        }
    }
}
```

b) (7 p)

```
public class Gui extends JFrame {
    private static int cellSize = 50;
    private GameEngine gameEngine;

    public Gui(GameEngine gameEngine, int size) {
        this.gameEngine = gameEngine;
        setTitle("The Maze Game");
        createCellViews(size);
        pack();
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setResizable(false);
        setVisible(true);
    }

    private void createCellViews(int size) {
        setLayout(new GridLayout(size, size));
        for (int i = 0; i < size*size; i++) {
            final int index = i;
            CellView cellView = new CellView(cellSize);
            cellView.addActionListener(
                new ActionListener() {
                    public void actionPerformed(ActionEvent e) {
                        gameEngine.makeVisible(index);
                    }
                });
            gameEngine.addCellObserver(index, cellView);
            add(cellView);
        }
    }
}
```

c) (1 p)

```
public class Main {
    public static void main(String[] arg) {
        private static int gameSize = 6;
        new Gui(new GameEngine(gameSize), gameSize);
    }
}
```