

PGR101 – Objektorientert programmering 2

7.6.17

Løsningsforslag

1a)

```
public Quiz() throws Exception {
    super("QUIZ");
    countCorrect = 0;
    count = 0;

    JPanel pnlWest = new JPanel(new GridLayout(3, 1));
    JPanel pnlEast = new JPanel(new GridLayout(3, 1));
    JPanel pnlSouth = new JPanel(new GridLayout(1, 2));

    txtQuestion = new JTextField(35);
    txtUserAnswer = new JTextField();
    txtUserCountCorrect = new JTextField();

    btnOK = new JButton("OK");
    btnExit = new JButton("Avslutt");
    btnOK.addActionListener(this);
    btnExit.addActionListener(this);

    pnlWest.add(new JLabel("Spørsmål"));
    pnlWest.add(new JLabel("Ditt alternativ"));
    pnlWest.add(new JLabel("Riktige"));

    pnlEast.add(txtQuestion);
    pnlEast.add(txtUserAnswer);
    pnlEast.add(txtUserCountCorrect);

    pnlSouth.add(btnOK);
    pnlSouth.add(btnExit);

    add(pnlWest, BorderLayout.WEST);
    add(pnlEast, BorderLayout.EAST);
    add(pnlSouth, BorderLayout.SOUTH);

    setVisible(true);
    pack();
    setDefaultCloseOperation(EXIT_ON_CLOSE);
    openFile();
    nextQuestion();
}
```

1b)

```
Object source = e.getSource();
if (source == btnOK) {
    char userChoice =
        txtUserAnswer.getText().charAt(0);
    if (userChoice == correctChoice) {
        ++countCorrect;
    }
    txtUserCountCorrect.setText("" + countCorrect
        + " av " + count + " riktige");
    nextQuestion();
}
```

1c)

```
} else
if (source == btnExit) {
    input.close();
    System.exit(0);
}
```

2a)

```
private void openFile() throws Exception {
    input = new Scanner(new File("quiz.txt"));
}
```

2b)

```
private void nextQuestion() {
    if (input.hasNext()) {
        txtQuestion.setText(input.nextLine());
        correctChoice = input.nextLine().charAt(0);
        count++;
    } else {
        txtQuestion.setText("Ikke flere spørsmål!");
    }
}
```

3a) Fra vedlegget om JFrame, ser vi arvesammenhengen mellom Object, Component, Container, Window, Frame og JFrame.

[java.lang.Object](#)
[java.awt.Component](#)
[java.awt.Container](#)
[java.awt.Window](#)
[java.awt.Frame](#)
[javax.swing.JFrame](#)

3b) En fordel man har ved å bruke arv, er at man kan legge til ny funksjonalitet i en klasse ved å arve fra ("utvide") en annen klasse, f.eks. vil JFrame i tillegg til egne egenskaper arve egenskapene i klassen Frame.

En annen fordel er at dette representerer "gjenbruk" – man slipper å skrive samme kode i flere klasser.

3c1) Forutsetter at getCredits og getSocialSecurityNumber finnes.
Linje 7 og 13 gir kompileringfeil.
Linje 7 – en subklassereferanse (s) kan ikke settes lik en superklassereferanse (o).
Linje 13 – klassen Object har ikke getFirstName()

3c2) Linje 7 og 13 kommenteres vekk.

3c3) (456) Ubi Dubius (123) 10
(456) Ubi Dubius (123) 10
(456) Ubi Dubius (123) 10
Ubi
Ubi

```
public class Client {
    public void main() {
        Student s =
            new Student("123", "456", "Ubi", "Dubius", 10);
        Person p = s;
        Object o = s;
        s = o;
        System.out.println(s.toString());
        System.out.println(p.toString());
        System.out.println(o.toString());
        System.out.println(s.getFirstName());
        System.out.println(p.getFirstName());
        System.out.println(o.getFirstName());
    }
}
```

Alle klassene har toString(), så linje 8 – 10 gir samme utskrift (kall på toString()).
getFirstName() ligger i Person (Student sin superklasse) – så linje 11 og 12 gir samme utskrift.