

# Forslag til løsninger til

## Skriftlig eksamen i Programmering og Udvidet Programmering

KVL, 22. december 1999

NB: Disse forslag til løsninger er ikke blevet tjekket på maskine. Løsningerne kan altså indeholde såvel syntaksfejl som andre fejl. Peter Sestoft 2000-01-04.

### Opgave 1

#### Opgave 1.1

Programmet udskriver

```
A B C
. . . 1
x x x 2
. . . 3
. . . 4
```

#### Opgave 1.2

```
void tjekOgReserver(int r, int s) throws Exception {
    if (ledig(r, s))
        reserver(r, s);
    else
        throw new Exception();
}
```

#### Opgave 1.3

```
int antalledige() {
    int antal = 0;
    for (int r=0; r<sæde.length; r++)
        for (int s=0; s<sæde[r].length; s++)
            if (sæde[r][s] == false)
                antal += 1;
    return antal;
}
```

## Opgave 2

### Opgave 2.1

Hvis der er mindst ét element, så skal alle de øvrige være lig det første element. Hvis der er 0 elementer, er de alle ens pr definition.

```
static boolean alleEns(int[] a) {
    if (a.length >= 1) {
        int a0 = a[0];
        for (int i=1; i<a.length; i++)
            if (a[i] != a0)
                return false;
    }
    return true;
}
```

Denne løsning er også i orden. Den er kortere, men til gengæld er den langsommere (fordi den laver dobbelt så mange tabelopslag):

```
static boolean alleEns(int[] a) {
    for (int i=1; i<a.length; i++)
        if (a[i-1] != a[i])
            return false;
    return true;
}
```

### Opgave 2.2

Der laves en heltalstabel fore med 14 pladser nummereret 0 til 13. For hver  $a[i]$  tælles den tilsvarende plads i fore op med 1, og tabellen fore returneres:

```
static int[] forekomster(int[] a) {
    int[] fore = new int[14]; // alle initialiseret til nul
    for (int i=0; i<a.length; i++)
        fore[a[i]] += 1
    return fore;
}
```

### Opgave 2.3

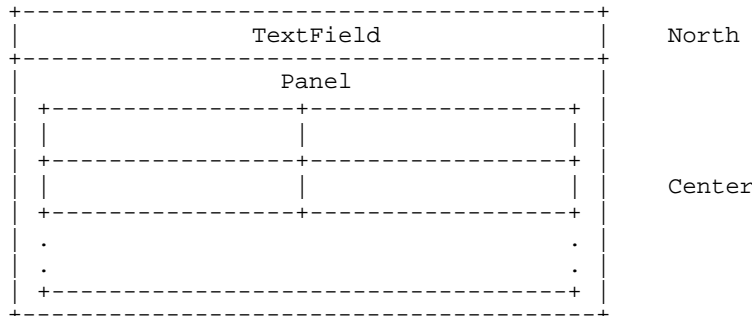
Hvis tabellen er tom, indeholder den 0 forskellige tal. Ellers indeholder den mindst ét tal. Fordi tabellen er sorteret står ens tal på samme sted i tabellen, så man kan nøjes med at tælle antal skift fra et tal til et andet:

```
static int antalforskellige(int[] a) {
    int antal = 0;
    if (a.length >= 1) {
        int senest = a[0];
        antal = 1;
        for (int i=1; i<a.length; i++)
            if (a[i] != senest) {
                senest = a[i];
                antal += 1;
            }
    }
    return antal;
}
```

## Opgave 3

### Opgave 3.1

Layoutet består af et ydre BorderLayout, der i North har et TextField og i Center har et Panel, der har GridLayout. Panelets GridLayout har 2 søjler og 10 rækker. I hver række er der først en Label og dernæst et TextField.



Ved punkt A i programmet skal man indsætte følgende erklæringer:

```
TextField karInd = new TextField();
TextField[] karUd = new TextField[10];
```

Ved punkt B i programmet skal man indsætte erklæringer og ordrer der opretter de nødvendige paneller, etiketter og tekstfelter og sætter layoutet sammen. Der laves først en tabel med de tekster der skal stå på etiketterne, sådan at man kan bruge en for-løkke til at sætte etiketter og tekstfelter på panelet p:

```
String[] indK = { "00", "03", "5", "6", "7", "8", "9", "10", "11", "13" };
setLayout(new BorderLayout());
add(karInd, "North");
Panel p = new Panel();
p.setLayout(new GridLayout(10, 2));
for (int i=0; i<10; i++) {
    p.add(new Label(indK[i]));
    karUd[i] = new TextField("0");
    p.add(karUd[i]);
}
add(p, "Center");
```

Alternativt er det også i orden at lave hver etiket og tekstfelt for sig. Det giver bare noget mere skrivearbejde:

```
setLayout(new BorderLayout());
add(karInd, "North");
Panel p = new Panel();
p.setLayout(new GridLayout(10, 2));
p.add(new Label("00")); karInd[0] = new TextField("0"); p.add(karInd[0]);
p.add(new Label("03")); karInd[1] = new TextField("0"); p.add(karInd[1]);
p.add(new Label("5")); karInd[2] = new TextField("0"); p.add(karInd[2]);
p.add(new Label("6")); karInd[3] = new TextField("0"); p.add(karInd[3]);
osv. for 7, 8, 9, 10, 11 og 13
add(p, "Center");
```

### Opgave 3.2

Ved punkt A i programmet skal man tilføje en erklæring af klassen FeltLytter. Klassens actionPerformed skal blot konvertere indholdet af tekstfeltet karInd til et tal og kalde den i opgaven omtalte metode optael:

```
class FeltLytter implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        optael(Integer.parseInt(karInd.getText()));
    }
}
```

Ved punkt B i programmet skal man indsætte en ordre der opretter et FeltLytter-objekt og knytter det til tekstfeltet karInd:

```
karInd.addActionListener(new FeltLytter());
```