Skriftlig eksamen

ITU, 20. januar 2000

NB: Ikke alle disse forslag til løsninger er blevet tjekket på maskine. Løsningerne kan altså indeholde såvel syntaksfejl som andre fejl. Peter Sestoft 2000-01-21.

Opgave 1

Opgave 1.1

Variablen `sum` er først 0, og tælles derefter op med 0, 1, 4, 9, 16 og 25, så programmet udskriver:

```
0
1
5
14
30
55
```

Opgave 1.2

Programmet udskriver 4 linier med 4 tegn, nemlig `o` når `(i+j)` er lige, og et `x` når `(i+j)` er ulige, altså:

```
oxox
xoxo
oxox
xoxo
```

Opgave 1.3

Figuren kan udskrives med dette program:

```
class Vedge {
  public static void main(String[] args) {
    for (int i=0; i<6; i++) {
      for (int j=0; j<i; j++)
        System.out.print(" ");
      System.out.print("*");
      for (int j=0; j<2*(6-i-1); j++)
        System.out.print(" ");
      System.out.println("*");
    }
  }
}
```

Den ydre løkke gennemløbes én gang for hver linie, med `i` lig 0, 1, 2, 3, 4, 5. På hver linie skrives `i` mellemrum, en stjerne, `2*(6-i-1)` mellemrum, og en stjerne.

Opgave 1.4

Når der er lavet en generel løsning på opgave 1.3 (som ovenfor), skal man blot indsætte for-løkkerne i en metodekrop og erstatte 6 med `n`, f.eks. sådan her:

```
static void kile(int n) {
  for (int i=0; i<n; i++) {
    for (int j=0; j<i; j++)
      System.out.print(" ");
    System.out.print("*");
    for (int j=0; j<2*(n-i-1); j++)
      System.out.print(" ");
    System.out.println("*");
  }
}
```

Opgave 2

Opgave 2.1

Metoden `maks` gennemløber tabellen `temp` og bruger den lokale variabel `maksimum` til at holde styr på den hidtil største temperatur:

```
static int maks(int[] temp) {
  int maksimum = 0;
  for (int i=0; i<temp.length; i++)
    if (temp[i] > maksimum)
      maksimum = temp[i];
  return maksimum;
}
```

Opgave 2.2

Metoden `antalfald` gennemløber tabellen og tæller variablen `fald` op med 1 hver gang en værdi i tabellen er større end den efterfølgende:

```
static int antalfald(int[] temp) {
  int fald = 0;
  for (int i=1; i<temp.length; i++)
    if (temp[i-1] > temp[i])
      fald++;
  return fald;
}
```

Bemærk at for-løkkens krop slet ikke udføres hvis der er 0 eller 1 tal i tabellen.

Opgave 2.3

Den ændrede metode `main` skal oprette en tabel der kan indeholde alle kommandolinieargumenterne, og konvertere hvert argument fra tegnstreng til tal, hvorefter `maks` og `antalfald` kaldes som før:

```
public static void main(String[] args) {
  int antal = args.length;
  int[] badetemp = new int[antal];
  for (int i=0; i<antal; i++)
    badetemp[i] = Integer.parseInt(args[i]);
  System.out.println("Maksimum: " + maks(badetemp));
  System.out.println("Antal fald: " + antalfald(badetemp));
}
```

Opgave 3

Opgave 3.1

Klassen Bygning kan erklæres sådan her:

```
class Bygning {
  int areal;
  int kvadratmeterpris;

  Bygning(int areal, int kvadratmeterpris)
  { this.areal = areal; this.kvadratmeterpris = kvadratmeterpris; }

  int vurdering()
  { return areal * kvadratmeterpris; }
}
```

Opgave 3.2

Klassen Grund kan erklæres sådan her:

```
class Grund {
  int areal;
  int kvadratmeterpris;
  int byggeret;

  Grund(int areal, int kvadratmeterpris, int byggeret) {
    this.areal = areal; this.kvadratmeterpris = kvadratmeterpris;
    this.byggeret = byggeret;
  }

  int vurdering()
  { return byggeret + areal * kvadratmeterpris; }
}
```

Selvom klassen Grund ligner Bygning meget, er det ikke særlig oplagt at lave den ene til en subklasse af den anden.

Opgave 3.3

Metoden `vurdering` skal gennemløbe tabellen `ejd` af ejendomme, kalde hver ejendoms metode `vurdering` og summere resultaterne:

```
static int vurdering(Ejendom[] ejd) {
  int sum = 0;
  for (int i=0; i<ejd.length; i=i+1)
    sum = sum + ejd[i].vurdering();
  return sum;
}
```

Opgave 3.4

Klassen ForurenetGrund kan erklæres sådan her:

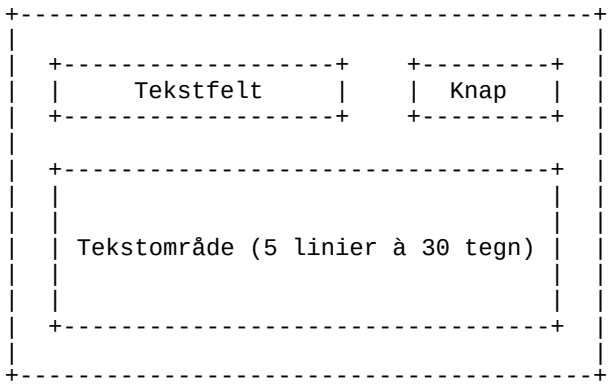
```
class ForurenetGrund extends Grund {
  int fradrag;

  ForurenetGrund(int areal, int kvadratmeterpris, int byggeret, int fradrag) {
    super(areal, kvadratmeterpris, byggeret);
    this.fradrag = fradrag;
  }

  int vurdering()
  { return super.vurdering() - fradrag; }

  void sætFradrag(int fradrag)
  { this.fradrag = fradrag; }
}
```

Bemærk at konstruktoren kalder superklassens konstruktor, og at `vurdering` kalder en metode i superklassen.

Opgave 4

Opgave 4.1

Appletten indeholder et enkelt panel, der ser omtrent sådan ud:



Appletten indeholder et enkelt panel, der har BorderLayout. Toppanelets West indeholder et tekstfelt (øverst til venstre). Toppanelets East indeholder et andet panel, der indeholder en knap mærket 'Læg til'. Toppanelets South indeholder et tekstområde med plads til 5 linier à 30 tegn.

Opgave 4.2

Efter handlingerne beskrevet i opgaven vil tekstområdet indeholde disse to linier:

```
Summen er nu 12
Summen er nu 20
```

Opgave 4.3

Den nye knap mærket 'Nulstil' kan sættes i Center på applettens toppanel. Ved `/* A */` tilføjes en erklæring af knappen:

```
Button nulstil = new Button("Nulstil");
```

Ved `/* B */` sættes knappen på toppanelet:

```
p.add(nulstil, "Center");
```

Ved `/* C */` knyttes en ny lytter til knappen:

```
nulstil.addActionListener(new NulstilLytter());
```

Ved `/* D */` erklæres lytterklassen NulstilLytter som en indre klasse:

```
class NulstilLytter implements ActionListener {
  public void actionPerformed(ActionEvent e) {
    sum = 0;
    ud.append("Sum nulstillet");
  }
}
```