

Danmarks Tekniske Universitet

Skriftlig prøve, den ... Test ...

Side 1 af 6 sider

Kursusnavn: Indledende Programmering og Softwareteknologi

Kursusnummer: 02100

Varighed: 2 timer

Tilladte hjælpemidler: Alle hjælpemidler er tilladt – uden adgang til internettet

Vægtningen af opgaverne er angivet ud for hver enkelt opgave.

Opgaverne besvares ved at aflevere en PDF-fil med løsningerne (programmerne og evt. forklaringer). Besvarelsen skal ikke indeholde programmer eller forklaringer, der allerede findes i opgaveteksten. Programmerne skal være formateret på sædvanlig måde, herunder indrykning af linjerne, men derudover stilles der ikke særlige krav til formateringen (det anbefales at benytte sort skrift på hvid baggrund).

Sammen med opgaverne udleveres en ZIP-fil `kode.zip` indeholdende de viste Java-klasser. Brug filerne i de tilsvarende opgaver, men filerne i den udleverede ZIP-fil `kode.zip` skal ikke afleveres.

Opgave 2 (Java) 20%

Skriv et Java-program Loop, hvor brugeren bliver bedt om at indtaste en række ord (streng) indtil ordet hedder STOP som i følgende eksempel:

```
Please enter "STOP" to stop.  
Enter a word: Betragt  
Enter a word: en  
Enter a word: sporvognsskinneskidtskraber  
Enter a word: STOP  
Here are the word lengths:  
Betragt 7  
en 2  
sporvognsskinneskidtskraber 27  
STOP 4  
The last word was "STOP" of course.
```

I eksemplet indtaster brugeren således ordene Betragt, en, sporvognsskinneskidtskraber og STOP.

Programmet skal beregne og udskrive længderne af de indtastede ord (jævnfør eksemplet).

Benyt et passende ArrayList-objekt til de indtastede ord.

Opgave 3 (Java) 40%

A)

Skriv en klasse `Data` (der benyttes i forbindelse med klassen `Grid` i den følgende delopgave).

Klassen skal have tre private felter `major`, `minor` og `active`, hvor de to første er heltal (typen `int`) og det sidste er en sandhedsværdi (typen `boolean`).

Klassen skal indeholde en konstruktør, der tager to heltal og en sandhedsværdi, således at felterne `major` og `minor` får heltalsværdierne i denne rækkefølge og feltet `active` får sandhedsværdien. For felterne `major` og `minor` skal et negativt heltal ændres til 0 (nul).

Derudover skal klassen indeholde følgende fem metoder.

Metoderne `getMajor`, `getMinor` og `isActive` tager ingen argumenter og returnerer henholdsvis værdierne i felterne `major`, `minor` og `active`.

Metoden `parity` tager ingen argumenter og returnerer ingen værdi. Værdierne i felterne `major` og `minor` opdateres til resten ved division med 2 (to).

Metoden `toString` tager ingen argumenter. En streng bestående af heltallet `major`, teksten `**` (to multiplikationstegn) og heltallet `minor` returneres, hvis sandhedsværdien `active` er sand, og ellers returneres en streng bestående af heltallet `major`, teksten `//` (to divisionstegn) og heltallet `minor`.

Der er ikke behov for flere metoder.

B)

Klassen `Data` skal benyttes i forbindelse med klassen `Grid` (jævnfør eksemplet herunder).

Betragt følgende Java-program:

```
public class Example {
    public static void main(String[] args) {
        boolean m[][] = { { true, false, true, true },
                           { true, false, true, false },
                           { true, false, true, true } };
        System.out.println("+++++++");
        Grid g = new Grid(m);
        System.out.println(g);
        System.out.println("+++++++");
        g.update();
        System.out.println(g);
        System.out.println("+++++++");
    }
}
```

Programmet skal udskrive følgende linjer på konsollen:

```
+++++++
0**0-0//1-0**2-0**3-1**0-1//1-1**2-1//3-2**0-2//1-2**2-2**3
+++++++
0**0-0//1-0**0-0**1-1**0-1//1-1**0-1//3-0**0-2//1-0**0-0**1
+++++++
```

Klassen `Grid` skal have et privat felt `a` af typen `Data[]` og må ikke have yderligere felter. Klassen skal også indeholde en konstruktør samt metoderne `update` og `toString`, så eksemplet giver den viste udskrift. Konstruktøren og metoderne beskrives i de følgende afsnit.

Konstruktøren tager et to-dimensionelt array af typen `boolean[][]`, og i det følgende kaldes parameteren `m` (for matrix). Konstruktøren skal oprette feltet `a`, således at elementerne svarer til elementerne i `m` rækkevis, og således at (række og søjle) indekserne i `m` og sandhedsværdien gemmes i det `Data`-objekt, der tildeles det pågældende array-element.

Metoden `update` returnerer ingen værdi. Metoden tager ingen argumenter, og for alle elementer i feltet `a`, hvor metoden `isActive` returnerer sand, så skal elementets metode `parity` kaldes.

Metoden `toString` returnerer en streng med alle elementerne i `a` adskilt af bindestreger (jævnfør eksemplet).

Eksemplets brug af `+++++++` er bare pynt.

Skriv klassen `Grid`.

Opgave 4 (Java) 25%

Skriv i alt 4 klasser uden `main` metoder således at klasserne `Vektor`, `Rektor` og `Lektor` nedarver direkte fra klassen `Person`.

Tilføj et privat felt `navn` af typen `String` til klassen `Person` og tilføj en konstruktør, der tager en streng som argument og initialiserer feltet med strengen. Tilføj desuden en `toString` metode uden argumenter, der returnerer feltets værdi.

Betragt følgende Java-program:

```
public class Main {
    public static void main(String[] args) {
        Vektor x = new Vektor("Kasper", 123456789);
        Rektor y = new Rektor("Jesper", 3.14);
        Lektor z = new Lektor("Jonathan");
        Person[] a = { x, y, z };
        printArray(a);
    }

    public static void printArray(Object[] a) {
        for (int i = 0; i < a.length; i++) {
            System.out.println(i + " " + a[i]);
        }
    }
}
```

Programmet skal udskrive følgende linjer på konsollen.

```
0 Vektor:Kasper;123456789
1 Rektor:Jesper;3.14
2 Lektor:Jonathan
```

Tilføj passende felter og konstruktører til klasserne `Vektor`, `Rektor` og `Lektor`. Tilføj desuden passende `toString` metoder uden argumenter.

Der skal ikke tilføjes yderligere felter, konstruktører eller metoder.