

02312 Indledende programmering, IMM DTU, Ver. 2009-09-09

Opgaver til 3. semester uge.

Anbefalet løsningsrækkefølge af opgaverne: 9, 12a 12b, 11, 13, 12c og 10.

Opgave 9. Test af logisk udtryk

Hvad er resultatet af følgende logiske udtryk?

Under forudsætning af at: $x = 2$ og $y = 4$.

Logisk udtryk	Værdi
<code>false</code>	false
<code>True</code>	true (må ikke stå med stort)
<code>true == false</code>	false
<code>x != y</code>	true
<code>x < 3 + y</code>	true
<code>y < x + 3</code>	true
<code>(x + y > 3) == false</code>	false
<code>false != x < 3</code>	true
<code>x == y == false</code>	true
<code>! false</code>	true
<code>! true</code>	false
<code>! true == false</code>	true
<code>! (true == false)</code>	true
<code>true && false</code>	false
<code>false true</code>	true
<code>x + y > 3 && x < y</code>	true
<code>x + y == 3 x < 4</code>	true
<code>x < y && (3*4 == 2*6-1*2+2) == !(3<x)</code>	false

Opgave 12a. Passwords igen

Ændre programmet fra opgave 8, så programmet bliver ved med at bede om et password, indtil brugeren har indtastet et korrekt et.

```
package opgave_12A;

import java.util.Scanner;

public class password {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        while (true){
            System.out.println("Indtast dit password:");
            String pass = scan.next();
            if(pass.length() < 4){
                System.out.println("Dit password er for kort! "
                    + "Det skal være mindst 4 tegn og dit er "
                    + pass.length());
            }else if(pass.length() > 9){
                System.out.println("Dit password er for langt! "
                    + "Det skal være højest 9 tegn og dit er "
                    + pass.length());
            }else{
                System.out.println("Dit password er accepteret. " +
                    "Det er " + pass.length() + " tegn langt");
                break;
                // return;
                // System.exit(0);
            }
        }
    }
}
```

Opgave 12b. ASCII tabel.

Skriv et program som beder brugerne om at indtaste et tegn.

Programmet skal herefter udskrive:

”Tegnet er et stort bogstav A-Z”

”Tegnet er et lille bogstav a-z”

”Tegnet er et tal 0-9”

”Tegnet er et af de 4 specialtegn (#, !, +, \)

Hvis tegnet ikke er indenfor de 4 ovennævnte grupper skriver

”Tegnet er ikke kendt af dette program.”

Hint: Brug deklarationen `char A = 'A'`, yderligere findes unicode tegntabel i lærebogens appendiks C.

Udvid programmet således at det ikke afslutter, men bliver ved med at spørge om nyt tegn. Find selv metode til at afslutte programmet.

```
package opgave_12B;

import java.util.Scanner;

public class ASCII {

    public static void main(String[] args) {
        while(true) {
            Scanner scan = new Scanner(System.in);
            System.out.println("Indtast et tegn (kun det første tegn gælder!) " +
                               "afslut med %");
            char c = scan.nextLine().charAt(0);

            if(c>='A' && c<='Z') {
                System.out.println("Tegnet er et stort bogstav A-Z");
            } else if(c>='a' && c<='z') {
                System.out.println("Tegnet er et lille bogstav a-z");
            } else if(c>='0' && c<='9') {
                System.out.println("Tegnet er et tal");
            } else if(c=='#' || c=='!' || c=='+' || c=='\\') {
                System.out.println("Tegnet er et af de 4 specialtegn " +
                                   "(#, !, +, \\)");
            } else if(c=='%') {
                System.exit(0);
            } else {
                System.out.println("Tegnets ASCII værdi er " + (int)c);
            }
        }
    }
}
```

Opgave 11. Datoer

Skriv et program, der indlæser en dato (tre heltal) i dette årtusinde og udskriver den næste dags dato.

Eksempel på kørsel:

Indtast år (mellem 2000 og 2999): 2001

Indtast måned: 9

Indtast dag: 30

Dagen efter 30/9 2001 er 1/10 2001

Hint: Indfør en variabel `daysinmonth` til at gemme antal dage i den indlæste måned.

Brug en switch-sætning til at initialisere `daysinmonth`.

Reglerne for skudår er som følger:

Hvis et årstal er deleligt med 4, er det et skudår. Dog ikke, hvis året også er deleligt med 100. Men alligevel, hvis det også er deleligt med 400.

Dvs. 1996 er et skudår, 2000 er et skudår, og 2100 er ikke et skudår.

```

package opgave_11;
import java.util.Scanner;
public class Dato {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        while (true){
            int year;
            do{
                System.out.println("Indtast et år mellem 2000 og 2999");
            }while ((year = scan.nextInt()) < 2000 || year > 2999);

            int month;
            do{
                System.out.println("Indtast en måned mellem 1 og 12");
            }while ((month = scan.nextInt()) < 1 || month > 12);

            boolean isLeapYear = false;
            if(year % 4 == 0){
                isLeapYear = true;
            }
            if(year % 100 == 0){
                isLeapYear = false;
            }
            if(year % 400 == 0){
                isLeapYear = true;
            }

            int daysInMonth;
            switch (month) {
                case 1:
                case 3:
                case 5:
                case 7:
                case 8:
                case 10:
                case 12:
                    daysInMonth = 31;
                    break;
                case 2:
                    if(isLeapYear){
                        daysInMonth = 29;
                    }else{
                        daysInMonth = 28;
                    }
                    break;
                default:
                    daysInMonth = 30;
            }

            int day;
            do{
                System.out.println("Indtast en dag mellem 1 og " + daysInMonth);
            }while ((day = scan.nextInt()) < 1 || day > daysInMonth);

            day++;
            if(day == daysInMonth + 1){
                day = 1;
                month++;
            }
            if(month == 13){
                month = 1;
                year++;
            }

            System.out.println("Dagen efter er: " + day + "/" + month + "-"
                               + year);
        }
    }
}

```

Opgave 13: Pyramider

Skriv et program, der udskriver pyramider:

```
*
*
***
*
***
*****
*
***
*****
*****
*
***
*****
*****
*****
```

Hint:

- Lav et for-loop til pyramider.
- Lav et for-loop til linier i pyramiden.
- Lav et for-loop til antal mellemrum.
- Lav et for-loop til antal "*" .

Det kan være nemmere at se løsningen, hvis du tegner den først.

```
package opgave_13;

public class Pyramids {
    final static int pyramidCount = 4;

    public static void main(String[] args) {
        for(int pyramid = 0; pyramid < pyramidCount; pyramid++){
            System.out.println("pyramid "+(pyramid+1));
            for(int line = 1; line <= pyramid+1; line++){
                String str = "";
                for(int i = 0; i < pyramid - line + 1; i++){
                    str += " ";
                }
                for(int i = 0; i < 2*line - 1; i++){
                    str += "*";
                }
                System.out.println(str);
            }
            System.out.println();
        }
    }
}
```

Opgave 12 C:

```
final int STOP=10;
int i;
for (i =1 ; i<STOP ; i++){
    System.out.print ("i har nu værdien " + i);
}
```

Ovenstående programstump anvender en for-løkke.

1) Lav et program som udfører det samme hvor ”for” løkken bliver erstattet af en af de 2 nedenstående:

- En do while
- En while sætning.

2) For de valgte løsninger kontrolleres det om disse gælder for alle værdier af stop.

3) Beskriv generelt hvordan en vilkårlig forsætning kan konverteres til de valgte typer af sætninger.

```
package opgave_12C;

public class NewLoops {
    final static int STOP=10;

    public static void main(String[] args) {
        int i;
        for (i =1 ; i<STOP ; i++){
            System.out.println("i har nu værdien " + i);
        }

        System.out.println();

        int j = 1;
        do{
            System.out.println("j har nu værdien "+j);
            j++;
        }while(j<STOP);

        System.out.println();

        int k = 1;
        while(k < STOP){
            System.out.println("k har nu værdien "+k);
            k++;
        }
    }
}
```

Opgave 10: Sinus funktion

Lav et program der kan:

- Plotte én sinus-periode med 20 "samples" på en standard konsol, der er 25 linier høj og 80 tegn bred . Brug kolonne 0-79 som $\sin(x)$ og linjenummer som x .
- Udvid programmet til at kunne tegne én sinus-periode bestående af et vilkårligt antal samples.
- Udvid programmet til at kunne tegne N perioder.
- Udvid programmet til at spørge om antal samples i perioden og gentage kørsel, indtil der svares med 0 eller negative tal.

Hint: Lad x-aksen være lodret og y-aksen vandret i konsollen

Nedenstående løsning er ikke komplet.

Udvidelserne kan implementere ved, at lade brugeren indtaste nogle parametre, som benyttes til at justere på de værdier, der her er konstanter.

```
package opgave_10;

public class VerticalSinus {
    final static int STEPS = 30;
    final static double STEPSIZE = Math.PI/8;
    final static int MULTIPLIER = 20;

    public static void main(String[] args) {
        for(int x = 0; x<STEPS; x++){
            int y = (int) (Math.sin(x*STEPSIZE)*MULTIPLIER);
            String str = "";
            for(int i = -MULTIPLIER; i<MULTIPLIER; i++){
                if(i == y){str+="*";}
                else if(i == 0){str+="|";}
                else{str+=" ";}
            }
            System.out.println(str);
        }
    }
}
```