Chapter 5: Conditionals and Loops

I dessa uppgifter kommer du att lära dig om hur man använder logiska satser och loopar för att lösa mer komplexa programmerings problem. Du kommer också att lära dig att strukturera program i separata moduler kallade funktioner (metoder).

Del 1: Övningar

Om du aldrig har programmerat är det obligatoriskt att du läser kapitlen 5 i boken samt skriver 1-2 programexempel från boken. Gör också nedanstående övning.

Activities at Lake LazyDays

As activity directory at Lake LazyDays Resort, it is your job to suggest appropriate activities to guests based on the weather:

```
temp >= 80:        swimming
60 <= temp < 80:   tennis
40 <= temp < 60:   golf
temp < 40:         skiing
```

1. Write a program that prompts the user for a temperature, then prints out the activity appropriate for that temperature. Use a cascading if, and be sure that your conditions are no more complex than necessary.

Del 2: Labbar

Läs boken och gör några övningar innan du börjar med labbarna.

BMI

Förbättra din BMI uppgift från labb kapitel 2 på följande sätt:

- Utöver vikt och längd skall programmet fråga om användarens kön samt ålder. Användaren skall mata in ”man” eller ”kvinna” samt sin ålder.
- Därefter skall programmet utifrån indata beräkna kroppsfett i procent enligt nedanstående formeln.

Man: Faktor= 16.2 Kvinna: Faktor = 5.4
*Fettprocent = (1.2 * BMI) + (0.23 * Ålder) – Faktor*

- Som slutmedelande till användaren skall programmet rekommendera motion eller inte beroende på det beräknade kroppsfett. Det normala kroppsfettet är mellan 12-20% för män och mellan 20-30% för kvinnor.

  OBS! Tänkt på att strängar jämförs med metoden equals se nedan:

```
if( text1.equals( text2 )){
 // gör något
}
```

PasswordCheck

Skriv ett program som läser in en sträng från kommandoraden och kontrollerar om det är ett "bra" lösenord. Ett "bra" lösenord betyder att det är:

- minst 8 tecken långt,
- innehåller minst en siffra 0-9,
- innehåller minst en stor bokstav,
- innehåller minst ett liten bokstav,
- innehåller minst ett icke-alfanumeriskt tecken.

Ett medelande till användaren skall skrivas.

PasswordMake

Skriv ett program som automatisk generarar password utifrån samma beskrivning som i förra uppgiften.

Encrypt

På föreläsningen har ni sett ett enkelt sätt att implementera en krypterings algoritm genom att manipulera ascii värdet för varje tecken.

```
import java.util.*;
    public class Encrypt
   {

       public static void main (String [] arg)
      {

         Scanner scan= new Scanner( System.in);
         System.out.println (" Write a text to enkrypt");
         String text =scan.nextLine();
         String crypt_text="";

         for(int i=0;i<text.length();i++){

            int ascii_ny=(int)text.charAt(i)+1;

            krypt_text=krypt_text+((char)ascii_ny);
         }
         System.out.println(krypt_text);
      }
   }
```

1. Gör om programmet genom att skriva krypteringsalgoritmen i en metod :
public static String encrypt(String klar_text) { …. }.
Metoden tar som input en sträng och returnerar den krypterade texten.
Använd nu metoden i programmet. Det för fungera precis som tidigare.

2. Skriv metoden
public static String decrypt(String crypt_text) { …. }
Metoden tar som argument en krypterad text och returnerar texten avkrypterad.
Använd metoden i ditt program.

PasswordCheck2

Gör om uppgifterna men flytta koden som undersöker password utanför main i en metod t.ex. kallad

public static boolean passwordCheck(String password) { }

Metoden skall returnera **true** om passworden är korrekt och **false** om inte.
Anropa sedan metoden i main. Skriv ut lämpligt meddelande beroende på vad metoden returnerar.

EmailCheck

Skriv en metod som undersöker om ett ord representerar en korrekt email adress eller inte.

public static boolean isEmail(String email) { }

Metoden skall returnera **true** om emailn är korrekt och **false** om inte.
En korrekt email innehåller bara bokstäver och siffror inga andra tecken utöver ”@ och .”
Tänk själv vilka andra villkor gäller för en korrekt email adress. Sätt de villkoren i din kod.

Testa metoden i ett program där du matar in alla möjliga email adresser. Fungerar det bra? Är det någon email adress som ditt program inte klarar av.

Ledning! Du kan lösa uppgiften på många sätt men kolla efter något som kallas ”Regular Expressions” eller ”Pattern”

LogInCheck

Gör motsvarade program men för login. Bestäm själv vad ett korrekt login är.

Calories

Förbättra också programmet Calories från kapitel 2.
Användaren skall få se en meny med aktiviteter och välja aktiviteteten. För varje aktivitet skall programmet fråga tiden man har utfört aktiviteten. När användaren väljer skall

programmet räkna antalet kalorier man har förbränt för respektive aktivitet samt addera det till det totala antalet. Här nedan se du en skalet av ett sådant program. Kopiera den och försök förstå strukturen. Gör den färdig. Lägg till några möjligheter till. T.ex, promenera, styrketräna.

```
// Kalorier / minut = 0,0175 * MET * vikt ( i kg) .


import java.util.Scanner;

  public class Calories
 {
   static final int METLöpning = 10;

    static Scanner scan = new Scanner(System.in);
    static int vikt;
    static double total_calories;


     public static void main(String[] args)
    {
      System.out.println( " Kaloriförbruknings program");
      System.out.println( " Hur mycket väger du ?");
      vikt = scan.nextInt();


       printMenu();
       int choice = scan.nextInt();

    // programmet stanar kvar i loopen tills användaren matar in 0
       while (choice != 0)
       {
         dispatch(choice);
          printMenu();
          choice = scan.nextInt();
       }


     } // main slutar här



    // Skriver ut användar möjligheter
        public static void printMenu()
    {
       System.out.println("\n  Välj aktivitet. Välj 4 när du är klar med alla val");
       System.out.println("   ==========");
       System.out.println("0: Exit ");
       System.out.println("1: Löpning" );
       System.out.println("2: Basket" );
       System.out.println("3: Sova");
       System.out.println("4: Beräkna kalorier");
```

```
        System.out.print("\nMata in dit val: ");
    }


  // metoden dispach behandlar och gör beräkningar beroende på val
     public static void dispatch(int choice)
    {

      switch(choice)
       {
         case 0:
               System.exit(0);
               break;
        case 1:
            System.out.println( " Hur lång tid har du sprungit ? Ange tiden i minut");
            int löpning_tid = scan.nextInt();
            total_calories = total_calories+ (0.0175 *vikt * METLöpning)*löpning_tid;

             break;

         case 2:
           break;

         case 3:
             break;
        case 4:

            System.out.println( " Förbrända kalorier är " + total_calories);
            System.out.println( " Beräkningen börjar på nytt");
            total_calories=0;
             break;



         default:
           System.out.println("Sorry, fel val");
      }
    }


}
```

FunnyGame

a) Hur många gissningar krävs det tills man träffar rätt då man måste hitta ett hemligt nummer mellan 1-N?
I värsta fall 100 men om man får lite hjälp i form av information om det gissade värdet är för stor eller för lite kan antalet försök minska väsentligt.

Skriv ett program som låter användaren mata in ett positivt nummer N och datorn skall sedan generera ett hemligt nummer mellan 1-N.
Därefter skall programmet låta användaren gissa talet tills hon träffar rätt. För varje försök skall datorn hjälpa användaren genom att medelande som ” FÖR LÅGT”, ” FÖR HÖGT”

Programet skall skriva ut hur många gissningar användaren gjorde tills hon träffade rätt.

Ledning: Använd en while konstruktion som loopar till användaren gissar rätt.

Rock, Paper, Scissor

Programmet Rock.java innehåller ett skelett för, spelet Sten, Påse, Sax (Rock, Paper, Scissor på engelska). Kopiera det och spara i din katalog.

Komplettera programmet så att det blir ett fullständigt spel dvs.

- Datorn kan inte slumpa bokstav, därför slumpar dator ett heltal 0-2. Därefter skall programmet i switch()göra om heltalet till en string som motsvarar ”R”, ”S” eller ”P”.
- Användaren matar in sitt val som en string ”R”, ”S” eller ”P”.

Programmet beslutar och skriver ut vinnaren enligt spelreglerna. Kan du inte spelreglerana Googla. Här nedan ser ett möjligt händelseupplopp när du kör programmet.
OBS! Tänk på att stor och små bokstäver är olika tecken. Välj att jobba med stora eller små men vara konsekvent.

Enter your play: R, P, or S
R
Computer play is S
Rock crushes scissors, you win!

```
import java.util.Scanner;
import java.util.Random;

public class Rock
{
    public static void main(String[] args)
    {
   String personPlay;     //User's play -- "R", "P", or "S"
   String computerPlay;   //Computer's play -- "R", "P", or "S"
   int computerInt;       //Randomly generated number used to
                          // determine computer's play

   Scanner scan = new Scanner(System.in);
   Random generator = new Random();

   //Get player's play -- note that this is stored as a string
   //Make player's play uppercase for ease of comparison
   //Generate computer's play (0,1,2)
   //Translate computer's randomly generated play to string
   switch (computerInt)
   {

   }

   //Print computer's play
   //See who won.  Use nested ifs instead of &&.
   if (personPlay.equals(computerPlay))
       System.out.println("It's a tie!");
   else if (personPlay.equals("R"))
       if (computerPlay.equals("S"))
      System.out.println("Rock crushes scissors.  You win!!");
       else

      //...  Fill in rest of code
    }
}
```

MyLibrary

Skapa en klass MyLibrary där du lägger en del användbara metoder.

a) Skriv i MyLibrary en metod
public static int maxOfTwo (int tal1, int tal2) {}.
Som du kan se den här metoden kommer att acceptera två heltal som parametrar och returnerar det största av dessa två.

b) I en annan klass som kallas TestUsefulMethods, skriv ett huvudprogram och testa dina maxOfTwo ().

För att göra det, först läs in två värden från användaren och sedan anropa maxOfTwo metoden med dessa två värden. Programmet skriver sedan ut det största.

c) Gå tillbaka till MyLibrary och implementera en metod public **static int maxOfThree (int tal1, int tal2, int tal3) {}.**

Metoden maxOfThree kommer att acceptera tre heltal som parametrar och ska returnera det största av dessa tre. Återanvänd metoden **maxOfTwo** i när du implementera **maxOfThre.**

d) Anropa denna metod i TestUsefulMethods i main().

Del 3: Extra (Ej obligatoriskt)

Date Validation

In this exercise you will write a program that checks to see if a date entered by the user is a valid date in the second millenium. A skeleton of the program is in *Dates.java*. Open this program and save it to your directory. As indicated by the comments in the program, fill in the following:

1. An assignment statement that sets monthValid to true if the month entered is between 1 and 12, inclusive.
2. An assignment statement that sets yearValid to true if the year is between 1000 and 1999, inclusive.
3. An assignment statement that sets leapYear to true if the year is a leap year. Here is the leap year rule (there's more to it than you may have thought!):

   > If the year is divisible by 4, it's a leap year UNLESS it's divisible by 100, in which case it's not a leap year UNLESS it's divisible by 400, in which case it is a leap year. If the year is not divisible by 4, it's not a leap year.

   Put another way, it's a leap year if a) it's divisible by 400, or b) it's divisible by 4 and it's *not* divisible by 100. So 1600 and 1512 are leap years, but 1700 and 1514 are not.

4. An if statement that determines the number of days in the month entered and stores that value in variable daysInMonth.

   If the month entered is not valid, daysInMonth should get 0. Note that to figure out the number of days in February you'll need to check if it's a leap year.
5. An assignment statement that sets dayValid to true if the day entered is legal for the given month and year.
6. If the month, day, and year entered are all valid, print "Date is valid" and indicate whether or not it is a leap year. If any of the items entered is not valid, just print "Date is not valid" without any comment on leap year.

Multiplication

Din lilla syster/bror ber dig att hjälpa henne/honom med multiplikation, och du bestämmer dig för att skriva ett Java-program som hjälper någon testa sina kunskaper.

Programmet kommer att låta henne välja ett startnummer, till exempel 5.
Därefter genererar programmet multiplikation frågor multiplikation frågor från 5 X 1 till 5x12.
För varje problem kommer hon att bli ombedd att ange det rätta svaret.
Programmet bör kontrollera svaret och inte låta henne vidare till rätt svar anges.
För att lämna programmet får användaren mata in -1 .