

DA339A Laboration L3

Syfte

Den här laborationens syfte är att du ska öva på att använda och skriva kod med variabler och konstanter och samt använda aritmetiska operatorer och viss strängmanipulation.

Observera: Lösningförslag och annan kod ges som bild med avsikten att du ska skriva om koden i stället för kopiering, vilket är ett effektivare sätt att bli mer uppmärksam, tänka över och förstå det du skriver.

Redovisning

Laboration L3 ska inte redovisas.

Förberedelser

Den här laborationen är direkt kopplad till det innehåll som finns i föreläsning F3 och tidigare laborationer.

Förbered en mapp för denna laboration. Skapa en ny katalog Lab3 på din dator för denna laboration. Du kan fortsätta som i tidigare labbar och skapa mappen där du har material från L1 och L2. Det som är viktigt är att du kommer ihåg och hitta katalogen när du vill kompilera via kommandetolken. Du bör också kunna lätt hitta dina filer när du vill jobba med dem nästa gång. Några exempel på en organiserad struktur:

- c/Users/AF6509/Kurser/DA339A/lab/lab3
- c/HT22/DA339A/lab/lab3
- m/Mau/DA339A/Lab/Lab3

Det är viktigt att skapa en bra katalogstruktur på din dator för att organisera dina kurser, terminer och deras innehåll. Spara ditt arbete ofta och ta alltid en backup av dina arbeten på till exempel ett USB-minne, en extern disk eller en annan plats på din dator. Detta är speciellt viktigt när du ska göra stora ändringar i ett tidigare projekt som fungerat bra.

Tips

Det är alltid bra att skriva några kommentarer och anteckningar för att komma ihåg sina tankar nästa gång man besöker koden. Det är också ett sätt att göra koden lättare att förstå för andra som granskar den.

- För kommentarer på slutet av en rad används två snedstreck (/). Kompilatorn hoppar då över allt du skriver från // till slutet av raden.
- Ett alternativt sätt är att använda /* ... */ om du vill skriva en kommentar över flera rader.
- Ta gärna för vana att skriva ditt namn i en kommentar längst upp i varje fil. Det underlättar på sikt och krävs när du ska lämna in kod i exempelvis inlämningsuppgifter.
- Indentera din kod med 1-tab (ofta 4 mellanslag) på varje nivå.

Uppgifter

Uppgift 1 - Räkna och jämföra tal

Öppna Atom (eller en annan textredigerare om du hellre vill det). Skapa en ny fil och spara den som *Calculator.java* i den mapp som du har skapat för Lab3.

Skriv följande kod i denna fil:

```
Calculator.java
1 // Calculator.java
2 // Sebastian Wöhrman, 2022-09-02
3
4 import java.util.Scanner;
5
6 /*
7  Denna klass har kod som utför olika typer av beräkningar
8  och omvandling mellan olika enheter.
9  */
10
11 public class Calculator
12 {
13
14     // I "metoden" main() startas vårt program, så kallad entry-point.
15     public static void main(String[] args)
16     {
17         System.out.println("Välkommen till världens bästa kalkylator!");
18
19     } // Slutet av main()-metoden
20
21 } // Slutet av Calculator klassen
22
```

Var noggrann och se till att alla semikolon, måsvingar, hakparenteser och parenteser hamnar på rätt ställe.

- ; semikolon
- {} måsvingar
- [] hakparenteser
- () parenteser

Öppna kommandotolken, navigera till den mapp där du sparat filen *Calculator.java*. Kompilera (javac) och kör (java) programmet för att kolla så allt fungerar. När programmet fungerar (som i figuren nedan), fortsätt med resten av uppgiften.

```
af6509@WC5CG1473GD7 MINGW64 ~/kurser/da339a/lab/lab3
$ javac Calculator.java

af6509@WC5CG1473GD7 MINGW64 ~/kurser/da339a/lab/lab3
$ java Calculator
Välkommen till världens bästa kalkylator!
```

Skriv den nya koden efter 'System.out.println("Välkommnen ...");' och innan den sista '}' som markerar slutet av main()-metoden. All kod skall alltså skrivas inom metodens start och slut (mer om metoder senare i kursen).

```

Calculator.java
3
4 import java.util.Scanner;
5
6 /*
7  Denna klass har kod som utför olika typer av beräkningar
8  och omvandling mellan olika enheter.
9  */
10
11 public class Calculator
12 {
13
14     // I "metoden" main() startas vårt program, så kallad entry-point.
15     public static void main(String[] args)
16     {
17         System.out.println("Välkommen till världens bästa kalkylator!");
18
19         /*Nedan undersöker vi hur resultatet av en division påverkas med hur
20         vi skriver ett numeriskt värde.*/
21         int tal1 = 9 / 5;
22         double tal2 = 9 / 5;
23         double tal3 = 9.0 / 5;
24         double tal4 = 9.0 / 5.0;
25
26         System.out.println("Tal 1, 9/5 ger: " + tal1);
27         System.out.println("Tal 2, 9/5 ger: " + tal2);
28         System.out.println("Tal 3, 9/5 ger: " + tal3);
29         System.out.println("Tal 4, 9/5 ger: " + tal4);
30
31     } // Slutet av main()-metoden
32
33 } // slutet av Calculator klassen
34

```

När du har skrivit koden färdigt, kompilera (javac) och kör (java) din applikation. Ifall du har felstavat något eller har andra syntax-fel, kommer kompileringen inte att lyckas. Läs felmeddelandet och försök att tänk över vad som kan ha orsakat felet, rätta det och kompilera igen. Om allt går bra kan du sedan köra programmet. Undersök resultatet och motivera varför resultat blir som det blir.

Ledning till resultatet finns under Lösningförslag sist i instruktionerna.

Som du kan notera i koden är ordningen på hur saker anropas (körs) och deklarerats (skapas) väldigt viktig. Prova att lägga raden 'double tal4 = 9.0 / 5.0;' sist i main()-metoden. Kompilera koden igen, vad händer? Rätta till din kod igen före nästa steg.

Uppgift 2A - Celsius till Fahrenheit

Vi ska nu skriva kod som ska fråga användaren om ett temperaturvärde i Celsius (det som vi använder i Sverige) och räkna om värdet till Fahrenheit grader. För att omvandla Celsius (C) till Fahrenheit (F), används följande formel:

$$F = 9/5 * C + 32$$

Börja med hjälp av Urforskaren/Finder att gå till katalogen där du sparat *Calcutalor.java*, kopiera java-filen och byt namn på kopian till *CalculatorCtoF.java*. Därefter öppna den nya filen i Atom och ändra dess klassnamn. Vi gör detta så att vi inte behöver börja skriva allt från början utan vi utgår från något befintligt och modifierar (rekommenderar att du tar bort de beräkningar som finns kvar från den gamla uppgiften men lämnar kvar ”System.out.println(Välkommen till...)” i den nya filen).

```

10
11 public class CalculatorCtoF
12 {
13
14     // I "metoden" main() startas vårt program, så kallad entry-point.
15     public static void main(String[] args)
16     {
17         System.out.println("Välkommen till världens bästa kalkylator!");

```

Vi ska göra tre varianter av denna uppgift.

1. Första varianten är att skriva Celsius värdet direkt i koden och beräkningen till Fahrenheit görs.
2. Den andra varianten är att vi får skriva in Celsius värdet med hjälp av en Scanner (se Lab2, Lab2Exercise2 programmet).
3. Den tredje varianten är att använda *JOptionPane* (annat sätt att skriva in värden i ett nytt fönster, se Lab2 *InputString* programmet) och *parseInt*.

Välj lämpliga datatyper för dina variabler och ge variablerna vettiga namn, inte bara F och C.

Här är ett körningsexempel:

Variant 1:

```

af6509@wC5CG1473GD7 MINGW64 ~/kurser/da339a/lab/lab3
$ java CalculatorCtoF
Välkommen till världens bästa kalkylator!
25.0C grader i Svergie motsvarar 77.0F grader i USA!

```

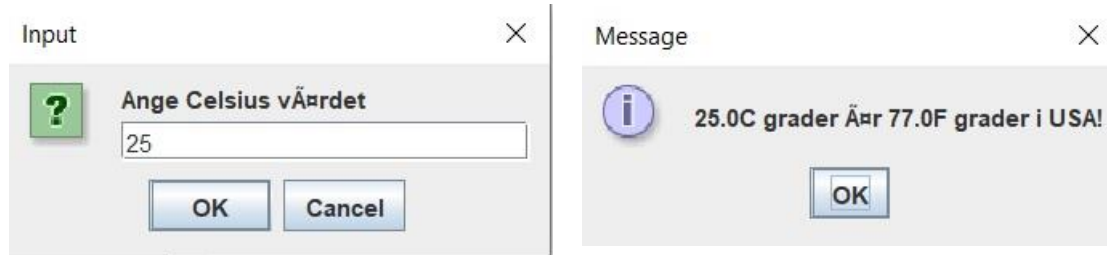
Variant 2:

```

af6509@wC5CG1473GD7 MINGW64 ~/kurser/da339a/lab/lab3
$ java CalculatorCtoF
Välkommen till världens bästa kalkylator!
Temperatur i C? 25
25.0C grader i Svergie motsvarar 77.0F grader i USA!

```

Variant 3:



Lösningförslag: finns i slutet av detta dokument.

Uppgift 2B Fahrenheit till Celsius

I och med att denna uppgift är lik uppgiften som du gjort rekommenderar vi att du gör som innan att du kopiera *CalculatorCtoF.java* och byter namn på kopian (till exempel *CalculatorFtoC.java*) samt att byta namn på klassen när du öppnar java-filen i Atom.

Skriv ny kod som liknar den läsning du har för Celsius till Fahrenheit men den ska i stället räkna om Fahrenheit (F) grader till Celsius (C). Använd följande formell:

$$C = 5/9 * (F - 32)$$

Testa med värdena: $F = 32 \rightarrow C = 0$, $F = 77 \rightarrow C = 25$

Denna uppgift ska använda samma tre varianter som användes i Uppgift 2A. Prova att skriva denna kod själv.

Uppgift 3 - Ticket

I denna uppgift skall du skriva ett nytt litet program som skriver ut kvitto på en biobiljett. Programmet skall fråga användaren om namn, antal vuxna och antal barn. En biobiljett kostar 100 kr för tillfället. Barn betalar alltid 75 % av detta belopp, dvs de får 25% rabatt. Programmet skall räkna ut det totala beloppet och skriva ut ett kvitto på det som bilden nedan visar.

Att göra:

- Skapa en ny java-fil och spara den som *Ticket.java*.
- Definiera klassen *Ticket*, och skriv stommen till *main()*-metoden.
- Skriv ut en hälsning från *main()*-metoden, kompilera och testa så allt fungerar (precis som vi gjorde innan) innan du går vidare.
- Barnrabatten på 25% (0.25) skall vara en konstant av typen *double*.

Obs. Du får använda svenska för variabelnamn, metodnamn och kommentarer, om det känns bättre.

Ett körnings-exempel:

```
af6509@WC5CG1473GD7 MINGW64 ~/kurser/da339a/lab/lab3
$ java Ticket

----- Välkommen till dagens familjefilm -----
----- Filmen kostar 100kr/person -----
----- Barn får alltid en rabatt på 25% -----

Ange ditt namn: Seb
Antal vuxna: 1
Antal barn: 1

Ditt kvitto: Seb
Totalt att betala: 175.0kr
Avdragen barnrabatt: 25.0kr

----- Välkommen åter! -----
```

Lösningförslag finns i slutet på detta dokument.

Uppgift 4

I denna uppgift kommer vi använda oss av *printf* (print formatting) samt *println*. Syftet är att vi vill få samma resultat på den output som visas i Git Bash oavsett vilken print-variant vi använder. Mer information om *printf* och *println* finns i länkarna nedan:

Printf:

<https://www.baeldung.com/java-printstream-printf>

Println:

<https://www.geeksforgeeks.org/system-out-println-in-java/>

String – Skriv först en *println* som skriver ut "Hello World!" i Git Bash. Därefter, med hjälp av länkarna ovan, skriv en *printf* som ger samma resultat.

Char – Skriv först en *println* som skriver ut en char 'm' och på nästa rad att du skriver en char 'M' i Git Bash. Därefter, med hjälp av länkarna ovan, skriv en *printf* som ger samma resultat.

Double – Skriv först en *println* som skriver ut en double 0.25 i Git Bash. Därefter, med hjälp av länkarna ovan, skriv en *printf* som ger samma resultat.

```
af6509@WC5CG1473GD7 MINGW64 ~/kurser/da339a/lab/lab3
$ java printfAndprintln
String:
Hello World!
Hello World!

Char:
m
M
m
M

Double:
0.25
0.25
```

Lösningförslag finns i slutet på detta dokument.

Uppgift 5

I denna uppgift ska vi demonstrera vad man får för felmeddelanden om man använder variabler av olika typ och inte typ-konverterar explicit samt vad som händer när man dividerar ett tal med 0 och om man försöker ändra en konstant.

Om en variabel-typ ska konverteras till någon annan kan man göra det explicit, dvs. att man skriver ut den nya typen inom parentes precis efter lika med tecknet (=). Typer kan konverteras hur som helst sinsemellan, utom i fallet boolean (sant/falskt). Boolean kan inte konverteras till eller från någon annan typ. Ett exempel på en explicit konvertering där vi konverterar ett decimaltal till ett heltal (int):

```
1 // Konvertering.java
2 // Sebastian Wöhrman, 2022-09-02
3 public class Konvertering
4 {
5     public static void main(String[] args)
6     {
7         System.out.println("Konvertering");
8         int tal1 = (int)9.5;
9
10        System.out.println("Tal 1 konverterat ger: " + tal1);
11    } // Slutet av main()-metoden
12 } // Slutet av Konvertering klassen
```

När vi kör programmet får vi följande resultat:

```
af6509@wC5CG1473GD7 MINGW64 ~/kurser/da339a/lab/lab3
$ java Konvertering
Konvertering
Tal 1 konverterat ger: 9
```

Här konverterades 9.5 till 9 och vi förlorade den data som kom efter decimaltecknet. Detta sker utan att kompilator klagar och det är viktigt att man tar hänsyn till detta så att man inte förlorar data som är viktigt för programmet.

Ladda ner Konvertering.zip från kurssidan, packa upp mappen och spara java-filen Konvertering.java i lab3 mappen du skapat för denna laboration. Därefter kompilera java-filen och granska felmeddelanden som visas.

```
af6509@wC5CG1473GD7 MINGW64 ~/kurser/da339a/lab/lab3
$ javac Konvertering.java
Konvertering.java:10: error: incompatible types: possible lossy conversion from
double to int
    int tal1 = 6.5;
               ^
Konvertering.java:11: error: incompatible types: possible lossy conversion from
double to int
    int tal2 = 1.5595;
               ^
2 errors
```

Öppna *Konvertering.java* i Atom och fixa konverterings felen, kompilera och kör programmet.

Nu när programmet går att köra igen ska vi skriva till kod som testar vad som händer när man försöker dividera ett tal med 0. Skriv kod som utför en division mellan ett tal och 0 (till exempel `int tal = 9/0;`) samt att variabeln är en `int` (heltal). Kompilera och kör programmet och se vad som händer. Testa nu att skriva samma uträkning fast med en `double` (till exempel `double tal = 9.0/0;`) och se vilken skillnad i meddelande när du kompilera och kör programmet igen.

Int tal = 9/0:

```
af6509@WC5CG1473GD7 MINGW64 ~/kurser/da339a/lab/lab3
$ java Konvertering
Konvertering
Exception in thread "main" java.lang.ArithmeticException: / by zero
    at Konvertering.main(Konvertering.java:16)
```

Double tal = 9.0/0:

```
af6509@WC5CG1473GD7 MINGW64 ~/kurser/da339a/lab/lab3
$ java Konvertering
Konvertering
Tal 4 (9.0/0): Infinity
Avangers: Infinity War!
```

Det sista vi ska kolla är vad som händer när man försöker ändra på en konstant. En konstant i programmering är en variabel som inte går att ändra värdet på efter den blivit tilldelat ett. Du kan använda samma java-fil för detta, deklarerar och initiera en konstant (man skriver det tagna ordet *final* före variabeltypen) längst ner i `main()`-metoden och på nästa rad försök ändra på konstantens värde.

```
21     final int tal5 = 100;
22     tal5 = 50;
```

Felmeddelandet som visas när du försöker kompilera säger att man inte kan ändra värdet på en konstant.

```
af6509@WC5CG1473GD7 MINGW64 ~/kurser/da339a/lab/lab3
$ javac Konvertering.java
Konvertering.java:23: error: cannot assign a value to final variable tal5
    tal5 = 50;
    ^
1 error
```

Vad är fördelen med att använda en konstant variabel? Tänk ut två olika situationer där en konstant variabel hade varit användbar.

Uppgift 6

I denna uppgift ska vi demonstrera vikten av parenteser vid beräkningar. Börja med att kopiera java-filen *Calculator.java* och byt namnet till *CalculatorParentes.java* (du kan såklart skapa en ny java-fil och skriva klassen och stommen till `main()`-metoden).

I `main()`-metoden ska du nu skriva två stycken beräkningar och därefter skriva kod så att resultaten visas i Git Bash. Du får använda vilka tal som helst, dock tänk på att använda lämpliga variabel-typer (`int`, `double` etc.).


```
int tal1 = 5 + 5 / 5; // räkning utan paranteser
int tal2 = (5 + 5) / 5; // räkning med paranteser
```

När du skrivit färdigt koden, kompilera och kör programmet. Om allt gått bra får du resultatet:

```
af6509@wC5CG1473GD7 MINGW64 ~/kurser/da339a/lab/lab3
$ java CalculatorParentes
Välkommen till världens bästa kalkylator!
Tal 1, ger: 6
Tal 2, ger: 2
```

I den första beräkningen (utan parenteser) tar divisionen prioritet och utförs först vilket ger resultatet 6. Medan i den andra beräkningen (med parenteser) utförs beräkningen inom parentesen innan divisionen utförs, vilket ger resultatet 2. Testa själv med andra värden och beräkningar och se hur mycket skillnad parenteser kan påverka resultatet.

Uppgift 7

I denna uppgift kommer du att använda modulus. Modulus kan beskrivas som ”resten” av en division när ett tal har blivit dividerat med ett annat tal. Använd länken nedan för en mer detaljerad beskrivning samt en modulus-kalkylator:

<https://www.calculatorsoup.com/calculators/math/modulo-calculator.php>

Din uppgift är att skapa en ny java-fil och ge den namnet *CalculatorModulus.java* (går bra att kopiera en av de tidigare *Calculator.java* programmen och byta namn). Därefter skriv ett program där användaren ska kunna mata in två tal att dividera och beräkna kvoten (resultatet när du dividerar ett tal med ett annat tal) och modulus (resultatet ska skrivas ut i Git Bash). Tänk på vilka variabel-typer du måste använda så att kvoten inte förlorar sina decimaler.

```
af6509@wC5CG1473GD7 MINGW64 ~/kurser/da339a/lab/lab3
$ java CalculatorModulus
Välkommen till världens bästa kalkylator!
----- Modulus + Kvot -----
Ange första talet: 9
Ange andra talet: 5
Kvoten: 1.8
Modulus: 4
```

Lösningsförslag finns i slutet på detta dokument.

Uppgift 8

Uppgift från kursboken. Deitel, sida 116, uppgift 2.15.

Write an application that asks the user to enter two integers, obtains them from the user and prints the square of each, the sum of their squares, and the difference of the squares (first number squared minus the second number squared).

Lösningsförslag

Uppgift 1

Tal 1: I denna uträkning blir svaret 1, detta beror på att variabeln *tal1* är en int (heltal). När uträkningen sker blir resultatet 1.8, dock när den konverteras till en int (heltal) avrundas svaret nedåt.

Tal 2: I denna uträkning är variabeln *tal2* deklarerad som en double (decimaltal), trots detta blir svaret 1.0. Detta beror på att inget av talen i uträkningen har någon decimal (9/5 och inte 9.0/5.0), dock skrivs decimaltal ut (1.0) för att *tal2* är en double. Vi kommer se skillnaden i resultaten av de sista två resultaten.

Tal 3: Som i *tal2* är variabeln *tal3* en double, dock är ett av talen skrivet med decimaltal (9.0) vilket gör att uträkningen tar hänsyn till antalet decimaler som ska vara med i resultatet vilket blir 1.8.

Tal 4: Samma som i *tal3* fast båda talen har en decimal i sig.

```
af6509@WC5CG1473GD7 MINGW64 ~/kurser/da339a/lab/lab3
$ javac Calculator.java

af6509@WC5CG1473GD7 MINGW64 ~/kurser/da339a/lab/lab3
$ java Calculator
Välkommen till världens bästa kalkylator!
Tal 1, 9/5 ger: 1
Tal 2, 9/5 ger: 1.0
Tal 3, 9/5 ger: 1.8
Tal 4, 9/5 ger: 1.8
```

Uppgift 2A

Skriv följande kod inne i klassen *CalculatorCtoF* (tre varianter). Koden för alla tre varianterna finns i samma java-fil, jag kommenterar bort de varianterna som jag inte vill kompilera (*/*...kod... */* kommentarer).

Variant 1:

```
20 // Variant 1 - Celsius till Fahrenheit
21
22 double celsius = 25.0;
23 double fahrenheit = (9.0/5.0 * celsius) + 32;
24 System.out.println(celsius + "C grader i Svergie motsvarar " +
25                     fahrenheit + "F grader i USA!");
26
```

Variant 2:

```
29 // Variant 2 - Celsius till Fahrenheit Scanner
30
31 System.out.print("Temperatur i C? ");
32 Scanner input = new Scanner(System.in);
33 double celsius = input.nextInt();
34 double fahrenheit = (9.0/5.0 * celsius) + 32.0;
35 System.out.println(celsius + "C grader i Svergie motsvarar " +
36                     fahrenheit + "F grader i USA!");
```

Variant 3:

```
40 // Variant 3 - Celsius till Fahrenheit JOptionPane och parseInt
41
42 String svar, celsiusString; // String lagrar ett antal tecken
43 celsiusString = JOptionPane.showInputDialog( "Ange Celsius värdet" );
44 double celsius = Integer.parseInt(celsiusString);
45 double fahrenheit = (9.0/5.0 * celsius) + 32;
46 svar = celsius + "C grader är " + fahrenheit + "F grader i USA!";
47 JOptionPane.showMessageDialog( null, svar);
```

Uppgift 2B

Ser nästan exakt ut som i 2A, bara beräkningen som ändras och att det är fahrenheit som är input samt att celsius är output.

Uppgift 3

Ticket kod (1/2):

```

1 // Ticket.java
2 // Sebastian Wöhrman, 2022-09-02
3
4 import java.util.Scanner;
5
6 public class Ticket
7 {
8     public static void main(String[] args)
9     {
10         // Skriv ut hälsningsmeddelanden
11         System.out.println();
12         System.out.println();
13
14         System.out.println("----- Välkommen till dagens familjefilm -----");
15         System.out.println("----- Filmen kostar 100kr/person -----");
16         System.out.println("----- Barn får alltid en rabatt på 25% -----");
17         System.out.println();
18
19         // Deklarera variabler
20         double barnRabatt = 0.25;
21         double pris = 100.0;
22         String namn;
23         int antalVuxna;
24         int antalBarn;
25         double totalKostnad;
26
27         // Ta input från kund/användare
28         Scanner input = new Scanner(System.in);
29         System.out.print("Ange ditt namn: ");
30         namn = input.nextLine();

```

Ticket kod (2/2), fortsättning:

```

33     System.out.print("Antal vuxna: ");
34     antalVuxna = input.nextInt();
35
36     System.out.print("Antal barn: ");
37     antalBarn = input.nextInt();
38
39     // Gör uträkningar
40     double rabatt = antalBarn * pris * barnRabatt;
41     totalKostnad = antalVuxna * pris + antalBarn * pris - rabatt;
42
43     // Visa resultatet och kvittot
44     System.out.println();
45     System.out.println("Ditt kvitto: " + namn);
46     System.out.println("Totalt att betala: " + totalKostnad + "kr");
47     System.out.println("Avdragen barnrabatt: " + rabatt + "kr");
48     System.out.println();
49     System.out.println("----- Välkommen åter! -----");
50 }
51
52 }

```

Uppgift 4

Kolla länkarna och läs om printf, finns många bra exempel.

```
1 // printfAndprintln.java, Sebastian Wöhrman, 2022-09-02
2 import java.util.Formatter;
3 import java.util.*;
4
5 public class printfAndprintln
6 {
7     public static void main(String[] args)
8     {
9         // String
10        System.out.println("String: ");
11        System.out.printf("Hello %s!\n", "World");
12        String string1 = "Hello World!";
13        System.out.println(string1);
14        // Char
15        System.out.println("Char: ");
16        System.out.printf("%c\n", 'm');
17        System.out.printf("%C\n", 'm');
18        char charLowerCase1 = 'm';
19        char charUpperCase1 = 'M';
20        System.out.println(charLowerCase1);
21        System.out.println(charUpp
22        // Double
23        System.out.println("Double: ");
24        System.out.printf("%3.2f\n", 0.25);
25        double double1 = 0.25;
26        System.out.println(double1);
27    } // Slutet av main()-metoden
28 } // Slutet av printfAndprintln klassen
```


Uppgift 6

Exempel på kod för uppgift 6 (beräkning med parenteser):

```
1 // CalculatorParentes.java
2 // Sebastian Wöhrman, 2022-09-02
3
4 /*
5  Denna klass har kod som utför olika typer av beräkningar
6  och visar vikten parenteser har vid de beräkningarna.
7  */
8
9 public class CalculatorParentes
10 {
11     // I "metoden" main() startas vårt program, så kallad entry-point.
12     public static void main(String[] args)
13     {
14         System.out.println("Välkommen till världens bästa kalkylator!");
15
16         int tal1 = 5 + 5 / 5; // räkning utan paranteser
17         int tal2 = (5 + 5) / 5; // räkning med paranteser
18
19         System.out.println("Tal 1, ger: " + tal1);
20         System.out.println("Tal 2, ger: " + tal2);
21     } // Slutet av main()-metoden
22
23 } // Slutet av CalculatorParentes klassen
```

Uppgift 7

Exempel på kod för uppgift 7 (beräkning av modulus och kvot):

```
1 // CalculatorModulus.java, Sebastian Wöhrman, 2022-09-02
2 import java.util.Scanner;
3
4 public class CalculatorModulus
5 {
6     public static void main(String[] args)
7     {
8         System.out.println("Välkommen till världens bästa kalkylator!");
9         System.out.println("----- Modulus + Kvot -----");
10        // Variabler som används vid beräkning.
11        int tal1;
12        int tal2;
13        double kvot;
14        int modulus;
15
16        Scanner input = new Scanner(System.in);
17        System.out.print("Ange första talet: ");
18        tal1 = input.nextInt(); // matar första värdet
19
20        System.out.print("Ange andra talet: ");
21        tal2 = input.nextInt(); // matar in andra värdet
22
23        kvot = (double)tal1 / (double)tal2; // Räknar ut kvoten
24        modulus = tal1 % tal2; // Räknar ut modulus
25
26        System.out.println("Kvoten: " + kvot);
27        System.out.println("Modulus: " + modulus);
28    } // Slutet av main()-metoden
29 } // Slutet av CalculatorModulus klassen
```