

## Laboration 12

Avsikten med laborationen är att ska skriva klasser som innehåller instansvariabler, instansmetoder och konstruktorer. Sedan ska du använda objekt av de klasser du skrivit i små program.

Skapa paketet **laboration12** innan du fortsätter med laborationen.

### Grundläggande uppgifter

#### Uppgift 12a

Denna uppgift bevarar du på papper.

Gör ett **klassdiagram** som beskriver klassen **Circle** nedan.

```
package laboration12;
```

```
public class Circle {  
    private double radius;  
  
    public Circle() {  
        this.radius = 1.0;  
    }  
  
    public Circle( double inRadius ) {  
        this.radius = inRadius;  
    }  
  
    public void setRadius( double inRadius ) {  
        this.radius = inRadius;  
    }  
  
    public double getRadius() {  
        return this.radius;  
    }  
  
    public double area() {  
        double area = Math.PI * this.radius * this.radius;  
        return area;  
    }  
  
    public String toString() {  
        return "CIRCLE, radius = " + this.radius;  
    }  
}
```

## Uppgift 12b

Nedanstående program använder klassen **Circle** i Uppgift 12a. Tyvärr innehåller programmet en del kod som ger kompileringsfel (understrukna). Kompileringsfelen beror på att instansvariabeln *radius* är *private*-deklarerad medan programmet försöker komma åt *radius* direkt.

Börja med att skapa klassen **Circle** och klassen **CircleTest** i paketet *laboration12*. Sedan ska du lösa problemet på två sätt, först ett dåligt sätt (1 nedan) och sedan ett bra sätt (2 nedan).

1. Lösning 1 – dålig lösning  
*public*-deklarera *radius* i klassen **Circle**. Nu går det bra att kompilera och exekvera **CircleTest**. Du har nu fått ett exempel på hur man kommer åt *public*-deklarerade instansvariabler. Sådana förekommer då och då.
2. Lösning 2 – bra lösning  
Deklarera åter *radius* som *private*. Ersätt samtliga understrukna delar i **CircleTest** med metoder. Det är alltid med metoder som du alltid ska kommunicera med objekt. Testkör programmet när du åtgärdat samtliga fel i programmet och kontrollera så att körresultatet är det tänkta.

```
package laboration12;
/**
 * Programmet använder ett objekt av typen Circle. Men på felaktigt sätt.
 * Rätta till felen i programmet.
 * @author Rolf Axelsson
 */
public class CircleTest {
    public static void main(String[] args) {
        Circle c = new Circle();

        c.radius = 23.2; // setRadius eller initiera via konstruktor
        System.out.println( "CIRKEL med radie = " + c.radius ); // getRadius
        System.out.println( "CIRKEL med area = " + Math.PI * c.radius * c.radius );
        // båda understrykningarna och lite till kan ersättas med area
        c.radius = c.radius + 5.7; // setRadius och getRadius, 1 el 2 rader med kod
        System.out.println( "CIRCLE, radius = " + c.radius ); // toString
    }
}
```

### Körresultat

```
CIRKEL med radie = 23.2
CIRKEL med area = 1690.9308298681701
CIRKEL med radie = 28.9
```

## Uppgift 12c

Du ska skapa en klass **Employee**. Klassdiagrammet till höger beskriver vad klassen ska innehålla. Diagrammet visar:

- Klassens namn
- Klassens attribut (instansvariabler). Minus-tecknet före namnet anger att variabeln ska vara *private*.
- Klassens konstruktorer och metoder (instansmetoder). Plus-tecknet före namnet anger att metoden ska vara *public*.

Hämta **EmployeeTest.java** från kurssidan och exekvera programmet.

**Var god vänd!**

Employee
<ul style="list-style-type: none"><li>- name : String</li><li>- employer : String</li><li>- wage : double</li></ul>
<ul style="list-style-type: none"><li>+ Employee( String, String, double)</li><li>+ setName( String )</li><li>+ setEmployer( String )</li><li>+ setWage( double )</li><li>+ getName() : String</li><li>+ getEmployer() : String</li><li>+ getWage() : double</li><li>+ toString() : String</li></ul>

Körresultatet ska bli:

Namn: Ergil Tuveberg

Arbetar hos: Renlighet AB

Lön: 27000.0

EMPLOYEE: name = Ergil Tuveberg, employer = Renlighet AB, wage = 27000.0

## Uppgift 12d

Klassdiagrammet till höger beskriver klassen

**BankAccount**. Skriv klassen BankAccount i paketet *laboration12*.

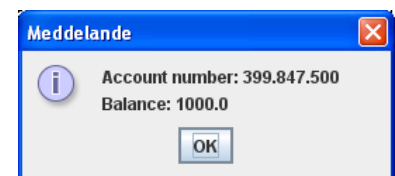
- **Konstruktorn med en parameter** ( String accountNbr ) initierar accountNbr till angivet värde. *balance* ska initieras till 0.0. *interestRate* ska initieras till 0.005.
- **Konstruktorn med tre parametrar** för över värdena till instansvariablerna.
- **get**-metoderna och **set**-metoden är som de brukar vara.
- Metoden **deposit** ska öka *balance* med parameterns värde. t.ex.

```
Konto konto = new Konto();  
konto.setSaldo( 1000 );  
konto.insättning( 1250 );  
System.out.println( konto.getSaldo() + " kr" );
```

ger utskriften: 2250.0 kr

- Metoden **withdrawal** ska på liknande sätt minska *balance* med parameterns värde.
- Metoden **info** ska visa ett fönster liknande det till höger.

Hämta **BankAccountTest.java** från hemsidan. Om du kör programmet ska du få följande resultat:



**Fråga 1:** Hur ändrar man innehållet i *accountNbr* i ett BankAccount-objekt? Ändringen ska ske från klassen BankAccountTest.

**Fråga 2:** Hur ändrar man innehållet i *balance* i ett BankAccount-objekt? Ändringen ska ske från klassen BankAccountTest.

## Uppgift 12e

Gör ett klassdiagram och skriv en klass som representerar ett bord (table).

Viktiga egenskaper för ett bord är *material* (*material*), *bredd* (*width*), *djup* (*depth*) och *höjd* (*height*). Välj själv lämpliga variabeltyper. Samtliga instansvariabler ska deklarerats som `private`

Viktiga metoder för ett bord är:

- *set-metoder* och *get-metoder* för samtliga egenskaper
- *size* vilken returnerar resultatet av ( bredd · djup )
- *toString* som returnerar en sträng på formen.  
"Table [ material = trä, width = 120, depth = 70, height = 72 ]"

## Uppgift 12f

Gör ett program som använder **Table**-klassen du skrivit i Uppgift 12e. Programmet ska anropa samtliga metoder i klassen Table.

## Uppgift 12g

I programmet **ContactTest** (se nedan) används ett objekt av typen **Contact**. Skriv klassen **Contact** så att programmet fungerar. Testkör programmet **ContactTest**. Fönsterna nedan visar sig om användaren matar in "Egon En", "040-123456", "0708-102030" och "egonen@hotmail.com":



```
package laboration12;
import javax.swing.*;

public class ContactTest {
    public void action() {
        Contact contact = new Contact();
        String name, phone;
        name = JOptionPane.showInputDialog("Ange namn");
        contact.setName(name);
        phone = JOptionPane.showInputDialog("Ange hemtelefon");
        contact.setPhone(phone);
        contact.setMobile(JOptionPane.showInputDialog("Ange mobil"));
        contact.setEmail(JOptionPane.showInputDialog("Ange mail-adress"));
        JOptionPane.showMessageDialog(null, contact.toString());
        JOptionPane.showMessageDialog(null, contact.getName() + "\n" +
            contact.getPhone() + "\n" +
            contact.getMobile() + "\n" +
            contact.getEmail());
    }

    public static void main(String[] args) {
        ContactTest prog = new ContactTest();
        prog.action();
    }
}
```

## Fördjupande uppgifter

### Uppgift 12h

Gör ett klassdiagram och skriv en klass som representerar ett hus (House).

Viktiga egenskaper för ett hus är *byggnadsår* (*year*), *bostadsyta* (*size*) och *tomtyta* (*garden*). Välj själv lämpliga variabeltyper. Samtliga instansvariabler ska deklarerars som *private*.

Klassen ska innehålla en konstruktor där användaren kan ange värden på samtliga egenskaper.

Viktiga metoder för en villa är:

- *set-metod* till bostadsyta och *get-metoder* för samtliga egenskaper
- *toString* som returnerar en sträng på formen.  
”Hus byggt 1972 med 142 kvm bostadsyta. Tomten är 620 kvm stor.”

Gör ett program som använder **House**-klassen du skrivit. Programmet ska anropa samtliga metoder i klassen House.

### Uppgift 12i

Inmatning från användaren via dialogfönster behövs i många program. T.ex. så kan inmatning av ett heltal ske via instruktionerna:

```
int res = Integer.MIN_VALUE;  
String str = JOptionPane.showInputDialog( "Ett meddelande" );  
if(str!=null && str.length()>0) { // användaren klickade inte på Avbryt  
    res = Integer.parseInt( str ); // och det finns inmatade tecken  
}
```

Instruktionerna ovan klarar av att användaren klickar på *Avbryt* och att användaren inte matar in några tecken innan klick på *OK*. Om något av detta inträffar så kommer *res* ha det minsta värdet som kan lagras i en *int*, nämligen konstanten *Integer.MIN\_VALUE*.

Instruktionerna ovan klarar däremot inte av om användaren matar in tecken som inte kan översättas till en *int*. Det ska vi åtgärda senare på kursen.

Placera klasserna **Input** respektive **InputTest** i paketet **laboration12**. I klassen **Input** ser du klassmetoden **getInt**, vilken låter användaren mata in ett heltal.

Studera *main*-metoden i klassen **InputTest** och exekvera sedan *main*-metoden.

Två dialoger låter användaren mata in totalt två heltal. Sedan skrivs de inmatade talen ut i en tredje dialog.



Testa programmet med olika inmatningar för att se vad som händer.

Lägg till klassmetoderna **getLong** respektive **getDouble** i klassen **Input**. Metoderna ska fungera på samma sätt som **getInt** men låta användaren mata in en *long* respektive en *double*. Testa metoderna så de fungerar bra.

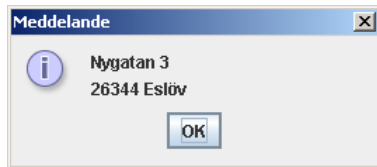
När du skriver **getLong** så kan du initiera *res* med konstanten *Long.MIN\_VALUE*.

När du skriver **getDouble** så kan du initiera *res* med konstanten *Double.NaN*.

## Uppgift 12j

Skriv klassen **Address** så den överensstämmer med klassdiagrammet till höger.

Kör programmet **AddressTest** och kontrollera att du får körresultatet nedan. Av programmet framgår bl.a. hur *toString*-metoden ska skrivas.



och därefter en dialog med den adress du matar in under körningen, t.ex.



Address
<ul style="list-style-type: none"><li>– street : String</li><li>– postalCode : int</li><li>– town : String</li></ul>
<ul style="list-style-type: none"><li>+ Address(String,int,String )</li><li>+ getStreet() : String</li><li>+ getPostalCode() : int</li><li>+ getTown() : String</li><li>+ toString() : String</li></ul>

## Uppgift 12k

**PP 5.6** Design and implement a class called **Box** that contains instance data that represents the height, width, and depth of the box. Also include a boolean variable called **full** as instance data that represents if the box is full or not. Define the **Box** constructor to accept and initialize the height, width, and depth of the box. Each newly created **Box** is empty (the constructor should initialize **full** to false). Include getter and setter methods for all instance data. Include a **toString** method that returns a one-line description of the box. Create a driver class called **BoxTest**, whose main method instantiates and updates several **Box** objects.

driver class – testklass, program som anropar olika metoder i en eller flera klasser.

## Extrauppgifter

### Uppgift 12m

**PaintWindow** (paketet laboration9) innehåller konstruktorn

```
public PaintWindow(ImageIcon icon)
```

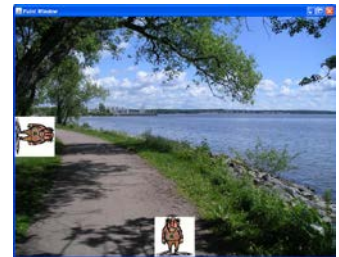
Om du använder konstruktorn så kommer den bifogade bilden bli bakgrundsbild på ritytan och bildens storlek bestämmer ritytans storlek (dock max 800x800 pixlar)

PaintWindow innehåller metoderna *getBackgroundWidth()* och *getBackgroundHeight()*. Metoderna returnerar ritytans bredd respektive höjd.

Plocka en bild från nätet och skriv ett program som:

- Skapar ett PaintWindow-objekt med bilden som bakgrund
- Visar bilden LiggandeGubbe.jpg så att gubben ”står” mitt på vänster fönsterkanten.
- Visar bilden Gubbe.jpg så att gubben står på mitten på nedre fönsterkanten.

Resultatet av programmet kan vara som fönstret till höger.



## Lösningsförslag

### Uppgift 12a

Circle
– radius : double
+Circle() +Circle( double )  + setRadius( double ) + getRadius() : double + area() : double + toString() : String

---

### Uppgift 12b

```
package laboration12;
import javax.swing.*;

public class CircleTest {
    public static void main(String[] args) {
        Circle c = new Circle( 23.2 );

        System.out.println( "CIRKEL med radie = " + c.getRadius() );
        System.out.println( "CIRKEL med area = " + c.area() );
        c.setRadius( c.getRadius() + 5.7 );
        System.out.println( c.toString() );
    }
}
```

---

### Uppgift 12c

```
package laboration12;

public class Employee {
    private String name;
    private String employer;
    private double wage;

    public Employee(String name, String employer, double wage) {
        this.name = name;
        this.employer = employer;
        this.wage = wage;
    }

    public void setName( String name ) {
        this.name = name;
    }

    public void setEmployer( String employer ) {
        this.employer = employer;
    }

    public void setWage( double wage ) {
```



```
        this.wage = wage;
    }

    public String getName() {
        return this.name;
    }

    public String getEmployer() {
        return this.employer;
    }

    public double getWage() {
        return this.wage;
    }

    public String toString() {
        return "EMPLOYEE: name = " + this.name + ", employer = " +
this.employer + ", wage = " + this.wage;
    }
}
-----
```

### Uppgift 12d

```
package laboration12;
import javax.swing.JOptionPane;

public class BankAccount {
    private String accountNbr;
    private double balance;
    private double interestRate;

    public BankAccount(String accountNbr) {
        this.accountNbr = accountNbr;
        this.interestRate = 0.005;
    }

    public BankAccount(String accountNbr, double balance, double
interestRate) {
        this.accountNbr = accountNbr;
        this.balance = balance;
        this.interestRate = interestRate;
    }

    public String getAccountNbr() {
        return accountNbr;
    }

    public double getBalance() {
        return balance;
    }

    public double getInterestRate() {
        return interestRate;
    }

    public void setInterestRate(double interestRate) {
        this.interestRate = interestRate;
    }

    public void deposit(double amount) {
        this.balance = this.balance + amount;
    }
}
```

```
public void withdrawal(double amount) {
    this.balance = this.balance - amount;
}

public void info() {
    String message = "Account number: " + this.accountNbr + "\n" +
        "Balance: " + this.balance;
    JOptionPane.showMessageDialog(null, message);
}
}
```

Fråga 1:

Det går inte att ändra *accountNbr* efter att objektet av typen *BankAccount* har skapats.

Fråga 2:

Man ändrar *balance* genom anrop till någon av metoderna *deposit* eller *withdrawal*.

-----

### Uppgift 12e

```
package laboration12;
```

```
public class Table {
    private String material;
    private double width;
    private double depth;
    private double height;

    public String getMaterial() {
        return this.material;
    }

    public void setMaterial(String inMaterial) {
        this.material = inMaterial;
    }

    public double getWidth() {
        return this.width;
    }

    public void setWidth(double inWidth) {
        this.width = inWidth;
    }

    public double getDepth() {
        return this.depth;
    }

    public void setDepth(double inDepth) {
        this.depth = inDepth;
    }

    public double getHeight() {
        return this.height;
    }

    public void setHeight(double inHeight) {
        this.height = inHeight;
    }

    public double size() {
        return this.width * this.depth;
    }
}
```

Table
– material : String – width : double – depth : double – height : double
+ setMaterial( String ) + getMaterial() : String + setWidth(double ) + getWidth() : double + setDepth(double ) + getDepth() : double + setHeight( double ) + getHeight : double + size() : double + toString() : String

```
    public String toString() {
        return "Table [ material = " + this.material + ", width = " +
            this.width + ", depth = " + this.depth + ", height = " +
            this.height + " ]";
    }
}
```

---

## Uppgift 12g

```
package laboration12;
```

```
/**
 * Klassen innehåller kontaktinformation till en person
 * @author Rolf
 */
public class Contact {
    private String name;
    private String phone;
    private String mobile;
    private String email;

    public void setName( String inName ) {
        this.name = inName;
    }

    public void setPhone( String inPhone ) {
        this.phone = inPhone;
    }

    public void setMobile( String inMobile ) {
        this.mobile = inMobile;
    }

    public void setEmail( String inEmail ) {
        this.email = inEmail;
    }

    public String getName() {
        return this.name;
    }

    public String getPhone() {
        return this.phone;
    }

    public String getMobile() {
        return this.mobile;
    }

    public String getEmail() {
        return this.email;
    }

    public String toString() {
        String res = "Namn: " + this.name +
            "\nHemtelefon: " + this.phone +
            "\nMobil: " + this.mobile +
            "\nEmail: " + this.email;

        return res;
    }
}
```

## Uppgift 12h

```
package laboration12;

public class House {
    private int year;
    private int size;
    private int garden;

    public House( int year, int sizeOfHouse, int sizeOfGarden ) {
        this.setYear(year);
        this.setSize(sizeOfHouse);
        this.setGarden(sizeOfGarden);
    }

    public void setSize(int sizeOfHouse) {
        this.size = sizeOfHouse;
    }

    public int getYear() {
        return this.year;
    }

    public int getSize() {
        return this.size;
    }

    public int getGarden() {
        return this.garden;
    }

    public String toString() {
        return "Hus byggt " + this.year + " med " + this.size + " kvm
bostadsyta. Tomten är " + this.garden + " kvm stor.";
    }
}
```

House
<ul style="list-style-type: none"><li>– year : int</li><li>– size : int</li><li>– garden : int</li></ul>
<ul style="list-style-type: none"><li>+ House( int, int, int )</li><li>+ setSize( int )</li><li>+ getYear() : int</li><li>+ getSize() : int</li><li>+ getGarden() : int</li><li>+ toString() : String</li></ul>

## Uppgift 12i

```
package laboration12;
import javax.swing.JOptionPane;

public class Input {
    public static int getInt( String message ) {
        int res = Integer.MIN_VALUE;
        String str = JOptionPane.showInputDialog( message );
        if(str!=null && str.length()>0) {
            res = Integer.parseInt( str );
        }
        return res;
    }

    public static long getlong( String message ) {
        long res = Long.MIN_VALUE;
        String str = JOptionPane.showInputDialog( message );
        if(str!=null && str.length()>0) {
            res = Long.parseLong( str );
        }
        return res;
    }

    public static double getDouble( String message ) {
        double res = Double.NaN;
        String str = JOptionPane.showInputDialog( message );
        if(str!=null && str.length()>0) {
            res = Double.parseDouble( str );
        }
    }
}
```

```
    }  
    return res;  
  }  
}
```

---

### Uppgift 12j

```
package laboration12;  
  
public class Address {  
    private String street;  
    private int postalCode;  
    private String town;  
  
    public Address(String street, int postalCode, String town) {  
        super();  
        this.street = street;  
        this.postalCode = postalCode;  
        this.town = town;  
    }  
    public String getStreet() {  
        return street;  
    }  
    public int getPostalCode() {  
        return postalCode;  
    }  
    public String getTown() {  
        return town;  
    }  
    public String toString() {  
        return this.street + "\n" + this.postalCode + " " + this.town;  
    }  
}
```

---

### Uppgift 12k

```
package laboration12;  
  
public class Box {  
    private double height;  
    private double width;  
    private double depth;  
    private boolean full;  
  
    public Box(double height, double width, double depth) {  
        this.height = height;  
        this.width = width;  
        this.depth = depth;  
        this.full = false;  
    }  
  
    public double getHeight() {  
        return height;  
    }  
  
    public void setHeight(double height) {  
        this.height = height;  
    }  
  
    public double getWidth() {  
        return width;  
    }  
}
```

```
    public void setWidth(double width) {
        this.width = width;
    }

    public double getDepth() {
        return depth;
    }

    public void setDepth(double depth) {
        this.depth = depth;
    }

    public String toString() {
        return "Box [width="+this.width+", depth="+this.depth+",
height="+this.height+", full="+this.full+"]";
    }
}
```

-----

### Uppgift 12m

```
package laboration12;
import laboration9.PaintWindow;
import javax.swing.ImageIcon;

public class Exercisel2m {
    public void action1() {
        ImageIcon image1 = new ImageIcon("C:/bilder/Sommar.jpg");
        ImageIcon image2 = new ImageIcon("C:/bilder/LiggandeGubbe.jpg");
        ImageIcon image3 = new ImageIcon("C:/bilder/Gubbe.jpg");
        PaintWindow window = new PaintWindow(image1);
        int width = window.getBackgroundWidth();
        int height = window.getBackgroundHeight();
        window.showImage(image3, (width-image3.getIconWidth())/2, height-
image3.getIconHeight());
        window.showImage(image2, 0, (height-image2.getIconHeight())/2);
    }

    public static void main(String[] args) {
        Exercisel2m ul2m = new Exercisel2m();
        ul2m.action1();
    }
}
```