

ITK:P1 Föreläsning 2

Introduktion till objektorientering

DSV Peter Mozelius

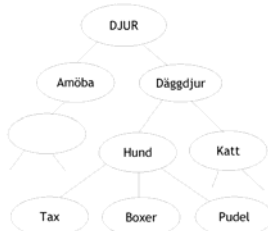
Objektorientering

- Världar uppbyggda av objekt
- Inte helt olik vår egen värld
- Ett sätt att *modularisera* våra system
- Objekten delas in i klasser
- Klasserna kan ärva varandra
- En klass blir en specialisering av en annan
- Ett antal klasser bildar en **taxonomi**

2

Objektorientering

En liten taxonomi



Klasserna ordnas
i en hierarki som
motsvarar deras
inbördes ordning

3

Objektorientering

- En klass innehåller en specifikation som definierar beteenden och egenskaper hos de klassinstanser (objekt) som man sedan skapar (instansierar) av klassen
- Av en klass Tax kan man sedan skapa en mängd olika taxart:

Tax t1 = new Tax();

4

Objektorientering

Klasser och klassinstanser

Klass

Katt

Instanser
(objekt)

Missan

Tigern

5

Objektorientering i Java

- En enkel klass med en konstruktor:

```
public class Katt {  
    private String namn;  
    public Katt(String s){  
        namn = s;  
    } //konstruktor  
} //klassen Katt
```

6

Objektorientering i Java

- I en annan klass kan vi sedan instansiera ett valfritt antal kattobjekt:

```
public class KattProgram1 {  
    public static void main(String[] arg){  
        Katt k1 = new Katt("Jameson");  
        Katt k2 = new Katt("Schrödinger");  
    } //main  
} //KattProgram
```

7

Objektorientering i Java

- För klasser, variabler och metoder finns det reserverade ord för åtkomstkontroll i Java
- De fyra val som finns är:
private
protected
public
ingen modifierare (paketåtkomst)

8

Programmering i Java

- Koden för ett KattProgram sparas i en fil i en katalog med namnet **KattProgram1.java**
- Källkoden kompileras sedan genom:
C:\katalog> **javac** KattProgram1.java
- Om koden är utan fel skapas en klassfil:
KattProgram1.class som kan exekveras med:
C:\katalog> **java** KattProgram1

9

Programmering i Java

Ett till litet Hello World:

```
public class HelloWorld {  
    public static void main(String[] arg){  
        System.out.print("Hello ");  
        System.out.println("World!");  
    }  
}
```

- I Javas klass **System** finns ett objekt **out** av typen **PrintStream** med metoder som **print()** och **println()**

10

Programmering i Java

- En objektorienterad variant av Hello World:

```
public class HelloPlace {  
    private String greeting = "Hello ";  
  
    public HelloPlace(String place){  
        greeting = greeting + place;  
    }  
  
    public String sayHello(){  
        return greeting;  
    }  
}
```

11

Programmering i Java

En klass som hållår ut vår hälsning:

```
public class HelloTest {  
    public static void main(String[] arg){  
        HelloPlace hp =  
            new HelloPlace("Stockholm");  
  
        String message = hp.sayHello() + "!";  
        System.out.println(message);  
    }  
}
```

12

Programmering i Java

```
public class Kattprogram2 {  
  
    public static void main(String[] args){  
        //Här anropas defaultkonstruktorn  
        Katt katt1 = new Katt();  
        System.out.println(katt1.visaKatt());  
  
        //Här anropas konstruktorn som tar ett argument  
        Katt katt2 = new Katt("Jansson");  
        System.out.println(katt2.visaKatt());  
    }  
}
```

13

Programmering i Java

```
class Katt{  
    private String kattvariabel="Det här är ";  
  
    public Katt(){  
        kattvariabel = kattvariabel + "en katt";  
    }  
    //defaultkonstruktör  
  
    public Katt(String namn){  
        kattvariabel = kattvariabel + namn;  
    }  
    //konstruktör som tar argument  
  
    public String visaKatt(){  
        return kattvariabel;  
    }  
    //visaKatt  
  
}
```

14

Jämförelseoperatorer i Java

- Vi behöver ibland jämförelseuttryck
- Jämförelseoperatorerna i Java är:
 == (lika med) != (inte lika med)
 < (mindre än) > (större än)
 <= (mindre än eller lika med)
 >= (större än eller lika med)
- Se upp med = (tilldelning) och == (jämförelse)

15

Logiska operatorer i Java

- Logiskt OCH: **&&**
- **(A && B)** är sant om både A OCH B är sanna

A	B	(A && B)
1	1	1
1	0	0
0	1	0
0	0	0

16

Logiska operatorer i Java

- Logiskt ELLER: **||**
- **(A || B)** är sant om A ELLER B är sann

A	B	(A B)
1	1	1
1	0	1
0	1	1
0	0	0

17

Logiska operatorer i Java

- Logiskt **ICKE** (invers) **!**
- OM boolean A = false;
- Så är **!A** sann
- **(A || !A)** är alltid sant
- **(A && !A)** är alltid falskt

Tack för idag!!

18
