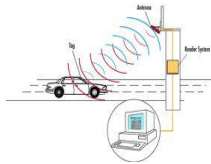


Algoritmer och datastrukturer

□ Allmänt om kursen

□ Kort javagrund repetition

- Klasser, metoder, objekt och referensvariabler,
- Hierarkiska klass strukturer
- Arrayer och arrayer av objekt
- Collection ramverket
 - ArrayList, HashMap



För utveckling av verksamhet, produkter och livskvalitet.



Vad handlar kursen om?

Mål: Att ge fördjupad kunskap om metoder och hjälpmedel för **analys** och **utformning** av **lösningar** av programmeringsproblem.

Detta omfattar :

- Att lära sig **implementera och tillämpa** några mycket kända **algoritmer och datastrukturer**
- Att lära sig göra **bedömningar av exekveringstiden för algoritmer**
- Att lära **TESTA** och göra **rimlighetsbedömningar**
- Att lära sig **programmera bättre**
- **Att ha roligt**

För utveckling av verksamhet, produkter och livskvalitet.



Vad handlar kursen om?

□ **Datastrukturer** - en samling av variabler av någon typ som hör ihop på något sätt och löser ett vist typ av problem på bästa sättet.

Ex: arrayer, listor, stack, träd, hashtabbel

□ Kända tillämpningar

Ex: räknemaskiner, kompilatorer, lexikon, simulatorer, internät och affärssystem applikationer

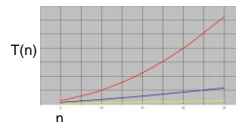
För utveckling av verksamhet, produkter och livskvalitet.



Vad handlar kursen om?

□ **Algoritmer** - metod för att stegvis lösa ett problem
Ex. Algoritm för effektiv sökning och sortering av data.

□ **Algoritm analys**, "space and time complexity" - hur olika implementationer påverkar exekveringen för en algoritm

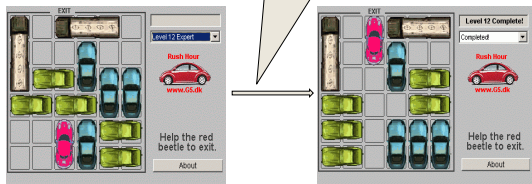


Exekveringstiden av en algoritm representeras som funktioner av inputdata

För utveckling av verksamhet, produkter och livskvalitet.



Varför?



Problem

Lösning

För utveckling av verksamhet, produkter och livskvalitet.



Kurs innehåll och planering

Webbsida : <http://www.hh.se/db4002>

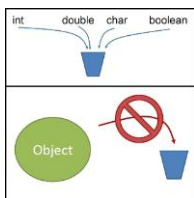


För utveckling av verksamhet, produkter och livskvalitet.



Java och andra språk

Primitiva datatyper



Definierade (egna) datatyper

För utveckling av verksamhet, produkter och livskvalitet.



Abstraktion

- Att *abstrahera* betyder att bortse från oväsentliga detaljer för att istället framhäva det som är relevant eller väsentligt.
- Vad som framhävs i en abstraktion kan vara olika från fall till fall beroende på i *vilket sammanhang* abstraktionen skall användas.
- När vi beskriver en viss del av verkligheten genom att abstrahera får vi en *abstrakt modell*.

För utveckling av verksamhet, produkter och livskvalitet.



Objekt-Orienterad Programmering

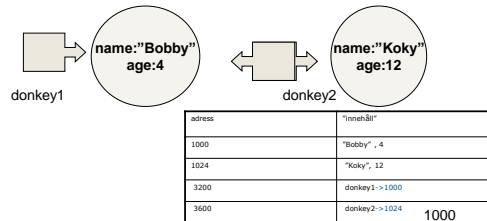
- **Klasser** är "modell" som beskriver hur objekt ska designas i termer av **egenskaper**, metoder och **konstruerare**.
- **Objekt** är *instance* av klassen, dvs "konkret data"



För utveckling av verksamhet, produkter och livskvalitet.



Objekt och referenser

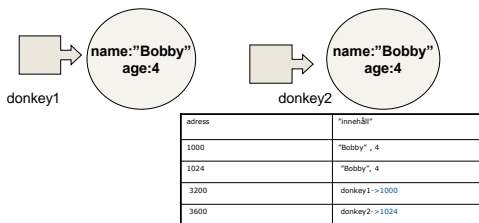


Vad händer om man gör : donkey2=donkey1?

För utveckling av verksamhet, produkter och livskvalitet.



Objekt och referenser



Vad blir if (donkey2==donkey1)?

För utveckling av verksamhet, produkter och livskvalitet.



Att jämföra objekt för "equality"

Om två objekt ska kunna jämföras måste de ha en metod som "gör detta".

Metoden skall heta equals().

```
class Program {
    public static void main(String[]
args) {
        String s1= "HBK är bäst" ;
        String s2="AIK är bäst";
        System.out.println(s1.equals(s2) );
    }
}
```

```
class Donkey{
    // data och metoder
    public boolean equals ( Object other)
    {
        ?????????????????????????
    }
}
```

[Klassen Object](#)

För utveckling av verksamhet, produkter och livskvalitet.



Objektorienterad mekanismen

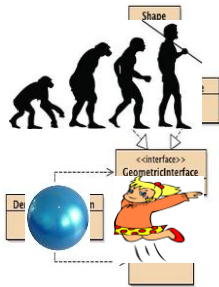
- Skapar återanvändbara hierarkier genom följande :

- **Inheritance(Arv)**

En klass kan arva från en annan (data och metoder) och utvidga dennes funktionalitet.

- **Interfaces**

En eller flera tomma metoder som **MÅSTE** implementeras av andra klasser för att garantera vissa egenskaper för objekten.



För utveckling av verksamhet, produkter och livskvalitet.



Skapa supertyper genom arv

```
class Person  
{
```

```
    private String name;  
    .....
```

```
    public Person( String n){  
        name=n; }  
}
```

```
    public String toString(){  
        return "The name is "+ name; }  
}
```



För utveckling av verksamhet, produkter och livskvalitet.



Interface Comparable

- I java biblioteket finns interfacen [Comparable](#) som används för att definiera objekt som kan jämföras och rangordnas

```
public interface Comparable  
{  
    public int compareTo( Object other );  
}
```

För utveckling av verksamhet, produkter och livskvalitet.



Interface och datatyp

```
import java.util.*;
```

```
class Person implements Comparable
```

```
{  
    // andra metoder
```

```
    public int compareTo(Object obj)
```

```
{  
    Person p=(Person) obj; // först casting från Object->Person  
    // kod som jämför
```

```
}
```

```
}
```

För utveckling av verksamhet, produkter och livskvalitet.



Super klass och subclass, **super**

class Student **extends** Person **implements** Comparable, Serializable {

```
private int credits;

public Student (int c, String n){
    super(n);
    credits=c;
}

public String toString() {
    return super.toString()+ " " + credits
}

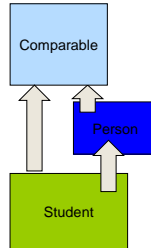
public int compareTo(Object other )
{
    // if credits ??
}
```



För utveckling av verksamhet, produkter och livskvalitet.



Super typ/ Sub typ



Person p1=new Person (" Kale Svensson");

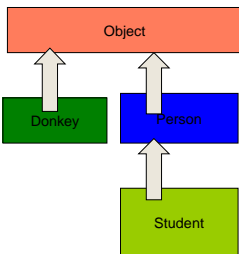
Student s1= new Student (0, " Tina Kaos");

Comparable obj = new Student (0, " Tina Kaos");

För utveckling av verksamhet, produkter och livskvalitet.



Datotypen Object är en "generell typ"



Vad blir utskriften ?
System.out.println(o1.toString());
System.out.println(o2.toString());

Object o1= new Person ("Kalle Svensson");

Object o2= new Student (0, "Anders T");

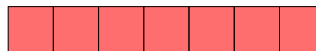
Object o3= new Donkey (" Bobby", 23);

För utveckling av verksamhet, produkter och livskvalitet.



Array av datotypen Object

Object [] array=new Object [7]



array[1]=new Person(..)
array[2]=new Donkey(...)

För utveckling av verksamhet, produkter och livskvalitet.



Datastrukturen array. Varför?

- Liknar den mest grundläggande databasen. Databas med en tabell.
- Data i arrayen skall vara av samma typ.
- När? Alltid mycket information skall lagras och manipuleras.
- Data i arrayen är lätt att nå

One Dimensional array

```
Initialization: int a[] = new int [12];
```

Value	1	2	3	4	5	6	7	8	9	10	11	12
Index	0	1	2	3	4	5	6	7	8	9	10	11

System.out.print(a[5]); Output: 6

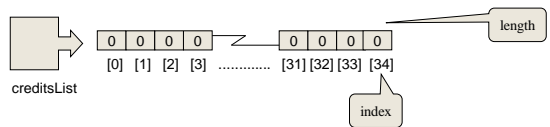
För utveckling av verksamhet, produkter och livskvalitet.



Datastrukturen array.

- Vi deklarerar och skapar array objekt

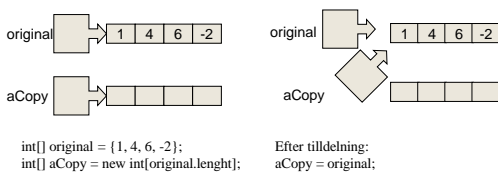
```
int[] creditsList= new int[35];
```



För utveckling av verksamhet, produkter och livskvalitet.



Datastrukturen array



- Detta fungerar inte, vi får två referenser till samma array

För utveckling av verksamhet, produkter och livskvalitet.



Arrayer som argument till metod

```
class A
{
    public static void förvirra( int [ ] a )
    {
        a[3]=7;
        a[4]=10;
        for(int i=0;i<a.length;i++)
            System.out.println(a[i]);
    }
}
```

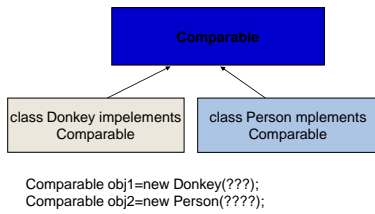
```
class Program {
    public static void main(String[] args) {
        int [] array1={5,6,7,8,3,8,5};
        A.förvirra(array1);

        for(int i=0;i<array1.length;i++)
            System.out.println(array1[i]);
    }
}
```

För utveckling av verksamhet, produkter och livskvalitet.



Interfacens namn blir datatyp, en **superdatatyp**



För utveckling av verksamhet, produkter och livskvalitet.



Klassen Object och Interface – är Javas sätt att implementera "generella" typer men....

```
class MyMethods
{
    public static void förvirra1( Object obj)
    {
        System.out.println(obj);
    }

    public static void förvirra2( Comparable obj1, Comparable obj2)
    {
        System.out.println(obj1.compareTo(obj2));
    }
}
```

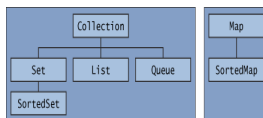
För utveckling av verksamhet, produkter och livskvalitet.



Collections

En datastruktur är:

- > data (representerad på olika sätt)
- > Metoder
- > Finns i java.util
<http://java.sun.com/j2se/1.5.0/docs/api/>



Collection- generell samling data (objekt)
List- ordnad samling av data
Set- ordnad samling av data som inte får innehålla dubletter
Map- Samling av data i par (objekt+ nyckel)
Queue- samling data som accessar efter en bestämd ordning

För utveckling av verksamhet, produkter och livskvalitet.

