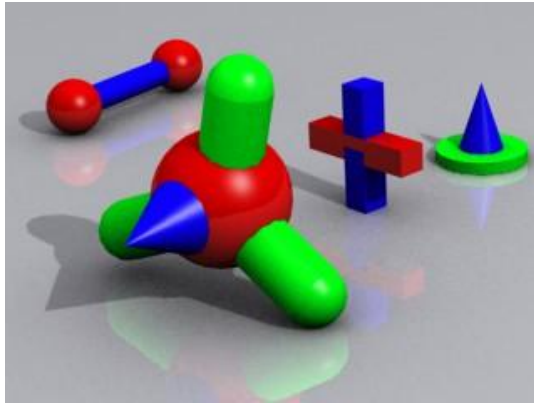


# ex5classes

Joachim von Hacht

# Flera men Olika



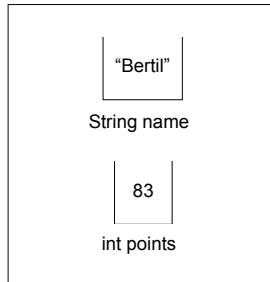
2

Arrayer används när vi behöver många variabler av samma typ.

Ibland behöver vi flera variabler men ev. av olika typ (vissa kan vara av samma typ)

- Om vi t.ex. vill beskriva en spelare i ett spelprogram så kanske vi behöver en variabel för spelarnamn och en för poäng
  - Krångligt t.ex. vid metodanrop att behöva skicka många variabler
- Istället för att ha enskilda variabler kan vi skapa ett **objekt** med flera variabler
  - Vi kan då skicka objektet, d.v.s. "alla" variablerna, på en gång till en metod.
- Dessutom hör ju de enskilda variablerna konceptuellt ihop, de används för att beskriva en spelare
  - Programmet blir lättare att förstå

# Konceptuell bild



Objekt med två  
variabler av olika typ

3

Ett objekt är en "förpackning" av variabler som hör ihop (mer senare)

- I bilden har vi samlat flera variabler som hör ihop med en spelare.

Vi kan inte skapa objekt direkt (förutom vissa specialfall, en sträng är t.ex. ett objekt)

- För att skapa objekt behöver vi en **klass**

OBS Bilden! Att värdet "Bertil" egentligen inte ligger i variabeln (mer senare)

# Klassdeklaration

```
// Very basic class
class Player {
    String name; // Instance variables
    int points;
}
```

4

För att skapa en klass i Java gör man en klassdeklaration

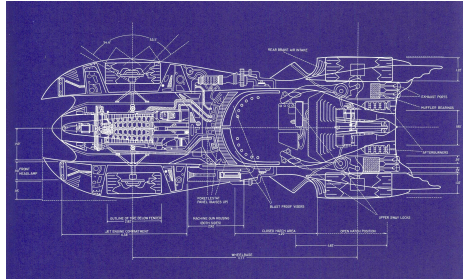
- Vi kan lägga klassdeklarationer var som helst i programmet.
- Vi lägger dem vanligen mot slutet i programmet (efter metoderna, inte i någon metod).
  - Eller i egna filer, se senare.

## Klassdeklaration

- Inleds med det reserverade ordet **class** + namnet på klassen.
  - Namn inleds med stor bokstav (CamelCase vid behov)
  - Namnet brukar vara ett substantiv och skall förklara vad objekten skall representera.
- Därefter ett block
  - I blocket deklarerar vi de variabler vi vill skall finnas i objekten, dessa kallas **instansvariabler** (alt. **attribut**)
  - I vilken ordning instansvariabler deklarerar spelar ingen roll (lite förenklat).

# Klass och Instans

Klass



Objekt (Instanser)



5

Vi kan skapa ett godtyckligt (ändligt) antal objekt utifrån en och samma klass

- Man kan se en klass som en ritning
- Vi säger att ett objekt är en **instans** av en klass
- Alla instanser får sin egen uppsättning av variabler
  - Vi kanske behöver två spelarobjekt, ett för "Pelle" och ett för "Fia", båda innehåller sina egna variabler name och points.

Analogier:

- Pannkaksrecept (klass) och pannkaka (objekt)
- Pepparkaksform (klass) och pepparkaka (objekt)

# Variabler för Objekt

```
// Declare, instantiate and initialize
Player p = new Player();
```

Ny typ!

Variabelnamn

new-operator

Metod med samma namn som klass

```
Player p1; // Declare ...
// ... Later instantiate and assign
p1 = new Player();
```

6

## En klass introducerar en ny typ, en **klasstyp**

- Eftersom klassen beskriver alla objekt, d.v.s. en mängd av objekt.
- Typsystemet är utbyggbart för klasser.
- Om vi skapar en ny klass skapar vi en ny typ! Vi kan vi deklarerar variabler av denna typ!
- När vi deklarerat variabeln kan vi initiera den
  - För att initiera variabeln måste vi instansiera (skapa) ett objekt.
  - Instansieringen sker genom att använda **new**-operatorn tillsammans med en metod som heter som klassen (som med arrayer)
  - Vi kan inte initiera som arrayer t.ex. { "olle", 87 } eller liknande går inte.

## Analys av kod (vänster till höger)

- Player är den nya typen som klassen introducerat
- p är namnet på variabeln
- new skapar ett objekt i minnet (inklusive de variabler som finns i objektet)
- Därefter körs metoden Player() för objektet, en sådan metod finns alltid men syns inte i koden, kallas för en konstruktor mer strax.
- Därefter initieras variabeln p med objektet (mer senare).

# Punktnotation

```
// Declare instantiate and initialize  
Player p1 = new Player();  
  
// Dot notation to access variables in object  
p1.name = "pelle";  
p1.points = 2;  
  
out.println( p1.name ); // "pelle"  
out.println( p1.points ); // 2
```

7

Ett namn på en array-variabel betecknade hela arrayen (lite förenklat, mer senare)

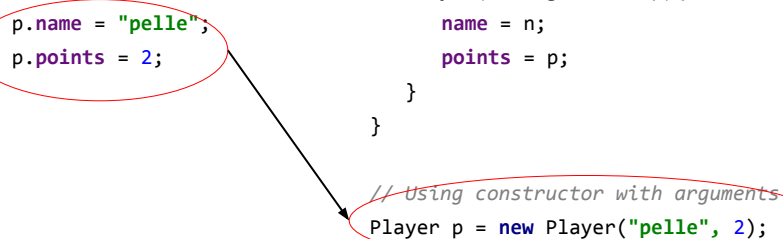
- För att komma åt enskilda variabler använde vi indexering

Ett namn på en objektvariabel (p1 i bilden) betecknar hela objektet (lite förenklat, mer senare)

- För att komma åt de enskilda variablerna används punktnotation d.v.s. objektvariabelnamn (p1), punkt, instansvariabelnamn (name)

# Konstruktör

```
class Player {  
    String name;  
    int points;  
}  
Player p = new Player();  
// Tiresome to set values  
p.name = "pelle";  
p.points = 2;  
  
// Better, use constructor  
class Player {  
    String name;  
    int points;  
// Constructor  
    Player(String n, int p){  
        name = n;  
        points = p;  
    }  
}  
  
// Using constructor with arguments  
Player p = new Player("pelle", 2);
```



8

För att initiera ett objekt på ett visst sätt kan vi skapa vi en speciell metod i klassdeklarationen, en **konstruktör**

- M.h.a. konstruktorn sätter vi värden för instansvariabler

En konstruktör

- Är en metod som automatiskt anropas i samband med instansiering (new)
  - Måste ha samma namn som klassen.
  - Parameterlista som vanlig metod
  - Får inte ange returtyp.
  - Kan inte anropas som en vanlig metod

Det finns alltid en parameterlös konstruktör (den vi använt tidigare).

- Om man skapar en konstruktör med parametrar försvinner den parameterlösa konstruktorn.
- Vill man ha en sådan får man skriva dit den själv.



# Arrayer med Objekt

```
// Declare and initialize array AND instantiate
// objects (each new .. creates an object)
Player[] players = { new Player("pelle", 23),
                     new Player("fia", 45) };

// First indexing then dot notation
out.println( players[0].name ); // "pelle"
out.println( players[1].points ); // 45
```

9

Om vi har en typ kan vi skapa arrayer utifrån typen

- Så också med klasstyper!
- För att komma åt enskilda variabler används ...
- ... först indexering eftersom vi började med en array (ger ett helt objekt) ...
- ... därefter punktnotation för att komma åt variabeln i objektet.

OBS! Bara `Player[] players = new Player[3];` skapar inte några player objekt, ... bara array-objektet skapas!