Objects First With Java
A Practical Introduction Using BlueJ

Att förstå klassdefinitioner

eller

Klasserna inifrån

Förra föreläsningen

- objekt
- klass
- metod
- parameter
- datatyp

En demonstration

- Projektet **picture**
- Projektet **lab-classes**

Exercises

1.30 Write the signature for a method named average that has two parameters, both of type int, and returns an int value.

1.31 Look at the book you are reading right now. Is it an object or a class? If it is a class, name some objects. If it is an object, name its class.

1.32 Can an object have several different classes? Discuss.

I dag

- fält
- konstruktorer
- metoder
- parametrar
- tilldelningssatser (tillordningssatser)
- villkorliga satser (kanske nästa gång)

Biljettmaskiner – utifrån

- Vi skall undersöka beteendet av en typisk biljettmaskin.
  - Projektet naive-ticket-machine används.
  - Maskiner ger biljetter med ett fixt pris.
    - Hur bestäms priset?
  - Hur sätter man in pengar?
  - Hur vet maskinen hur mycket pengar satts in?

Biljettmaskiner - inifrån

- Interaktionen med ett objekt tillåter oss gissa (delar av) objektets beteende.
- När vi analyserar klassen ser vi HUR beteendet uppstår.
- Alla Java-klasser har en likadan intern struktur.

Grundstruktur

```
public class TicketMachine
{
    Klassens inre del borttagen.
}
```

Yttre "förpackning" av TicketMachine

```
public class KlassNamn
{
    Fält
    Konstruktorer
    Metoder
}
```

Innehåll av en klass

Fält

- Fält lagrar värde för ett objekt.
- Kännda också som instansvariabler.
- Använd Inspect i BlueJ för att undersöka fält.
- Fält definierar objektets tillstånd.

```
public class TicketMachine
{
    private int price;
    private int balance;
    private int total;

    Konstruktorer och metoder
      finns ej med.
}
```

visibility modifier type variable name

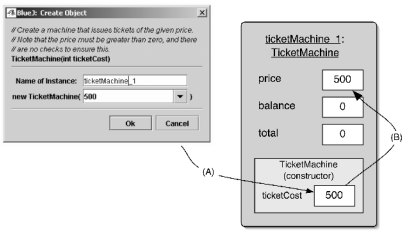
```
private int price;
```


Konstruktorer

- Konstruktorerna skapar objekt.
- De har samma namn som deras klass.
- De lagrar initiala fältvärden.
- Ofta används parametrarna för att föra in externa värde här.

```
public TicketMachine(int ticketCost)
{
    price = ticketCost;
    balance = 0;
    total = 0;
}
```


Parametrarna används för att flytta värde



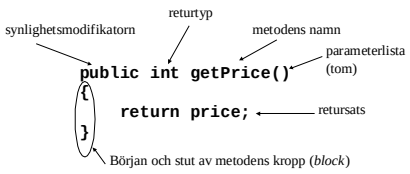
Tilldelning

- Värde lagras i fält (och i andra variabler) m.h.a. tilldelningssatser:
  - variable = expression;
  - `price = ticketCost;`
- En (enkel) variabel lagrar bara ett värde: det gamla värdet försvinner efter tilldelningen.

Accessmetoder

- Åtkomstmetoder, accessorer
- Metoder implementerar objekts beteende.
- Accessorer ger information om objektet.
- Alla metoder har liknande struktur: ett huvud och en kropp.
- Huvudet definierar metodens signatur.
  `public int getPrice()`
- Kroppen innehåller alla metodens satser.

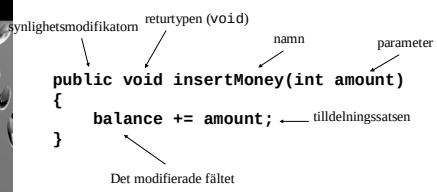
Accessorer



Muteringsmetoder

- Har samma struktur: huvudet och kroppen.
- Används för att mutera (d.v.s. ändra) objektets tillstånd.
- Detta görs genom att ändra värde i ett eller flera fält.
  - Typiskt innehåller tilldelningssatser.
  - Typiskt tar parametrar.

Muteringsmetoder



Att skriva ut från metoder

```
public void printTicket()
{
    // Simulate the printing of a ticket.
    System.out.println("##################");
    System.out.println("# The BlueJ Line");
    System.out.println("# Ticket");
    System.out.println("# " + price + " cents.");
    System.out.println("##################");
    System.out.println();

    // Update the total collected with the balance.
    total += balance;
    // Clear the balance.
    balance = 0;
}
```

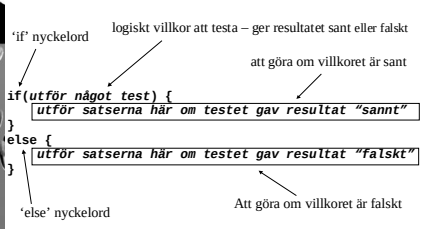

Problem hos TicketMachine

- Det görs ingen kontroll om värdet på amount är negativt!
- Ingen kontroll görs om kunden har tillräckligt mycket pengar för biljetten!
- Det betalas inte tillbaka pengar om kunden lägger i för mycket!
- Det görs ingen kontroll om biljettpriset som ges som en parameter till konstruktorn är negativt!
- Kan den förbättras?
  Kör better-ticket-machine:-)

Att välja

```
public void insertMoney(int amount)
{
    if(amount > 0) {
        balance += amount;
    }
    else {
        System.out.println("Use a positive amount: " +
                           amount);
    }
}
```

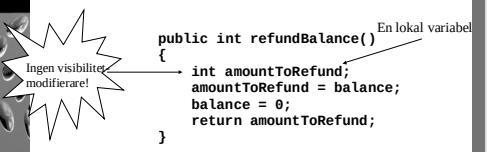

Att välja



Fält och lokala variabler

- Fält
  - Lagrar värden som finns kvar under objektets livstid.
  - Är åtkomliga från alla delar av klassen.
- Metoder kan innehålla lokala variabler
  - Lagrar värden som bara finns medan metoden exekveras.
  - Är bara åtkomliga inuti metoden.

Lokala variabler




```
public int refundBalance()
{
    int amountToRefund;
    amountToRefund = balance;
    balance = 0;
    return amountToRefund;
}
```


Översikt

- Klassernas kroppar inehåller fält, konstruktorer och metoder.
- Fält lagrar värde som bestämmer objektens tillstånd.
- Konstruktorer skapar objekt.
- Metoder implementerar objektens beteende.

Översikt

- Fält, parametrar och lokala variabler är variabler.
- Fält finns hela objektslivstiden.
- Parametrarna används för att ta emot värde i konstruktorer eller metoder.
- Lokala variabler används för temporär lagring av information.

Översikt

- Objekt kan ta beslut m.h.a. villkorssatser (if).
- Ett test med resultat “sant” eller “falskt” tillåter metoder att välja en av två alternativa sätt att agera.