

## Objektorienterad programmering

- En föreläsning om:
  - Objektorienterad programmering
    - Dess mål och metoder
    - Vad är ett objekt?
  - Avvägningar vid design av objektorienterade språk
  - Stöd för objektorientering i ert favoritspråk

2000-11-03

Thomas Johansson Datavetenskap

1

## Systemutveckling

- Brister, missförstånd ger
  - - Fel system
  - - Fel tidpunkt
  - - Fel kostnad

2000-11-03

Thomas Johansson Datavetenskap

2

## Objekt-orientering i tre steg

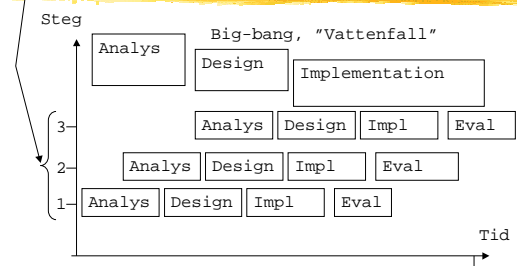
- OOA; Analys av verkligheten
- OOD; Design av implementation
- OOP; impl mha OO-programmspråk

2000-11-03

Thomas Johansson Datavetenskap

3

## Inkrementell systemutveckling



2000-11-03

Thomas Johansson Datavetenskap

4

## Dokument - Beskriver

- |                  |  |
|------------------|--|
| ■ Projektplan    | ■ Hur arbetet ska utföras, tidplan, vad som ska produceras, granskning |
| ■ Kravspec       | ■ Slut användarens behov   |
| ■ Analys         | ■ OO modell över verkliga obj  |
| ■ Design         | ■ Systemarkitektur, relationer, beteende                               |
| ■ Detaljdesign   | ■ Beskriver varje klass "manual för klasserna"                         |
| ■ Källkod        | ■ Hur systemet ska testas  |
| ■ Testprotokoll  |  |
| ■ Användarmanual |  |

2000-11-03

Thomas Johansson Datavetenskap

5

## Målen med objektorientering

- Modellering - nära verkligheten; lättförstått
- Modularisering - kring verkliga begrepp; stabil
- Förändring - enkel lokalisering
- Utbyggnad - nya klasser, operationer el. arv
- Återanvändbarhet - mha arv
- Produktivitet - klassbibliotek
- Terminologi - samma i alla faser; spårbarhet

2000-11-03

Thomas Johansson Datavetenskap

6

## Mål forts

- Att skapa en OO-modell som underlag
- Skapa öppen arkitektur "infrastruktur"
- Maximera inkapsling & modularitet
- Minimera beroenden
- Studera objektens samarbete

2000-11-03

Thomas Johansson Datavetenskap

7

## Början ...

- Historik
  - Simula 67
  - Smalltalk 80
- Procedurorienterad programmering
  - Subprogram
  - Programbibliotek
- Dataorienterad programmering
  - Abstrakta datatyper
  - Objektbaserade språk, föregångare till...

2000-11-03

Thomas Johansson Datavetenskap

8

## Objektorienterade programmeringsspråk

- Smalltalk, Eiffel och Java är rena objektorienterade språk
- C++ och Ada stöder objektorientering
- Det går att programmera objektorienterat i C eller assembler för den delen
  - Språket måste stödja någon form av inkapsling,
  - T.ex. på fil-nivå

2000-11-03

Thomas Johansson Datavetenskap

9

## Den objekt-orienterade modellen

- Beskriver **vad** som agerar (objekten)
- Gemensamt språk för alla
  - Intressenter
  - Nivåer
  - Tidpunkter
- Hög nivå utan detaljer

2000-11-03

Thomas Johansson Datavetenskap

10

## Jämför med icke-OO

- Procedurella språk/metoder
- Beskriver hur **något** utförs
- Data ligger ofta centralt och alla rutiner kan komma åt det
- Programflödet är det viktiga

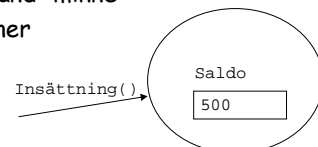
2000-11-03

Thomas Johansson Datavetenskap

11

## Vad är ett objekt?

- VAD vi hanterar i vårt system
- Har tillstånd "minne" mitt\_konto
- Operationer
- Identitet



2000-11-03

Thomas Johansson Datavetenskap

12

## Relationer

- Klass-Objekt
- Användning
- Association
- Aggregat
- Arv

## Krav

- Tre krav måste vara uppfyllda:
  - Abstrakt datatyp
  - Arv
  - Polymorfism och dynamisk bindning
- Den abstrakta datatypen kallas ofta klass
  - Data kallas **attribut**, funktionaliteten finns i **metoder**
  - Anrop av en metod kallas att skicka ett **meddelande** till ett objekt

## Inkapsling

- Inkapsling brukar också tas upp som ett krav på ett objektorienterat språk
- C++ har tre nivåer av inkapsling
  - private
  - protected
  - public
- Java har även 'package scope'

## Tillbaka till 80-talet ...

- Problem med objektbaserade språk
  - Abstrakta datatyperna borde kunna återanvändas
  - Alla typdefinitionerna oberoende och på samma nivå
- Lösningen heter Arv
  - Data och funktionalitet kan ärvas och modifieras
  - Återanvändning utan ändringar i redan skriven kod

## Återanvändning

- Inom systemet
- Av generella bibliotek
- Applikationsspecifika
- Väldefinierade klasser
- Köpa färdigt
- Inom koncernen

## Problem med återanvändning

- Målet ej uttalat av projektledningen
  - Avsätt resurser i planeringen
- Socialt/psykologiskt
  - Snäva tidsramar - ej generell kod
  - Använder ej gärna annans kod
- Administrativt
  - Insamling, kvalitetsgranskning
  - Dokumentera, katalogisera, distribuera
  - Söka & finna

## Arv ...

- En subclass ärver från en basclass (kallas också superklass)
  - Åtkomst av basclassens attribut och metoder kan styras
  - Metoder kan omdefinieras (override)
  - Två sorters metoder och två sorters attribut
    - Instansmetoder och attribut
    - Klassmetoder och attribut

2000-11-03

Thomas Johansson Datavetenskap

19

## Computing with an Object-Oriented Language

- Ett objektorienterat program som körs kan ses som en samling datorer/objekt som kommunicerar via meddelanden.
- Varje objekt är en abstraktion av en dator i den meningen att det lagrar data och kan manipulera det.
- Objektorienterad programmering är att lösa problem genom att identifiera objekten i problemet och sedan simulera dem, dess processer och den nödvändiga kommunikationen.

2000-11-03

Thomas Johansson Datavetenskap

20

## Avvägningar vid design av ett objektorienterat språk

- Tre sätt att angripa problemet med typsystemet:
  - Den 'rena' modellen:
    - Allt är objekt, från det minsta heltal till det största datasystem
    - + Elegant form
    - - De enklaste operationer måste ske genom metदानrop
  - Utgå från ett imperativt språk och lägg till en objektmodell
    - + Snabbare
    - - Mycket större, mer komplext system

2000-11-03

Thomas Johansson Datavetenskap

21

## Tre sätt ...

- Objektmodell i botten men med 'imperativ stil' för primitiva datatyper
  - + snabbt för heltal och andra primitiva typer
  - - Komplikationer när metoder bara kan ta objekt som parametrar -> Wrapper Classes
- C++, Java hör till den senare typen
- Allt är objekt utom de primitiva datatyperna som int, double, ...
- String och fält [] är objekt i Java

2000-11-03

Thomas Johansson Datavetenskap

22

## Statisk eller dynamisk typkontroll ?

- Om språket skall vara starkt typat:
  - -> Statisk typkontroll
  - Restriktioner vid polymorfism
  - Två sorters typkontroll vid metदानrop:
    - Parametrarna måste stämma
    - Returtypen likaså
  - Kanske tillåtet med 'assignment compatibility'
- Alternativet är dynamisk typkontroll
  - Vänta med kontrollen tills metoden anropas
  - Kostsamt och försenar typkontrollen

2000-11-03

Thomas Johansson Datavetenskap

23

## Allokering/deallokering av objekt

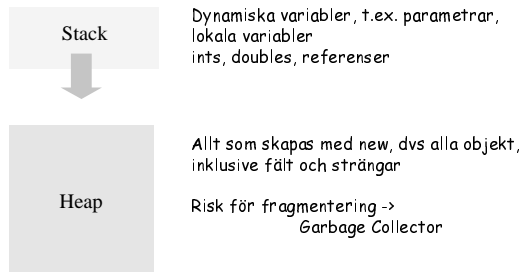
- Var allokeras objekten ? Tre sätt :
  - Statiskt av kompilatorn
  - Dynamiska objekt på stacken
  - Eller på heapen med new
- Om new det enda sättet -> uniform metod
  - inga pekare som måste derefereras a la C++
- C++ har alla sätten - och alla problemen
- Java har bara new (allt på heapen)

2000-11-03

Thomas Johansson Datavetenskap

24

## Stack och heap



## Vi använder new, sen då ?

- Hur deallokera objekten ?
- -> Explicit
  - Problem med 'dangling pointers'
  - C++ har **delete**
- -> Implicit
  - Vi behöver *Garbage Collection* i någon form
  - Java har *GC*