

# 1 Object Oriented Programming

DAT042  
D2 Ip1 2011/2012

Uno Holmer  
chalmers@unoholmer.se

1

## Course Contents

- Introduction to object-oriented programming...
- ...with a strong software engineering foundation...
- ...aimed at producing and maintaining large, high-quality software systems

Object oriented programming, DAT042, D2, 11/12, Ip 1

Lecture 1 2

## Buzzwords

responsibility-driven design  
inheritance      overriding      encapsulation  
iterators      javadoc      interface      coupling  
cohesion      refactoring      abstraction  
collection classes      mutator methods  
regression testing      design patterns  
polymorphic method calls

Object oriented programming, DAT042, D2, 11/12, Ip 1

Lecture 1 3

## Goals

- Sound knowledge of programming principles
- Sound knowledge of object-orientation
- Able to critically assess the quality of a (small) software system
- Able to implement a small software system in Java

Object oriented programming, DAT042, D2, 11/12, Ip 1

Lecture 1 4

## Recommended book

Jan Skansholm

*Java direkt*

Studentlitteratur

*... or any equivalent book*

Object oriented programming, DAT042, D2, 11/12, Ip 1

Lecture 1 5

## Course overview (1)

- Objects and classes
- Class definitions
- Object interaction and references
- Grouping objects, lists, arrays, iterators
- Program documentation
- Library classes
- Well-behaved objects - testing, maintaining, debugging

Object oriented programming, DAT042, D2, 11/12, Ip 1

Lecture 1 6

## Course overview (2)

- Inheritance
- Polymorphism
- Graphical user interfaces
- Error handling with exceptions
- Streams and files
- Concurrent programming with active objects and threads
- Application design and design patterns
- Modelling with UML

Object oriented programming, DAT042, D2, 11/12, lp 1

Lecture 1 7

## Activities

- Lectures
- Supervised class excercises
- Laborations (4)
- Your own work (plenty)

Object oriented programming, DAT042, D2, 11/12, lp 1

Lecture 1 8

V.	Dag	Nr	Föreläsning	Litteratur
35	mån	1	Introduktion till kursen och OOP	S 1
	tis	2	Objekt och klass, instansvariabler, klassvariabler, metoder	S2-3
	tor	3	Objektinteraktion, referenser, listor	S2.5, S4.5.1-2, S9.7
	fre	4	Fält, iteration, iteratorer	S3.8-9, S9.1-3, S9.8-9, S17.4-5
36	mån	5	Associativa samlingar, konstanter	S17.7.1-3, S3.3, S2.3
	tis	6	Programdokumentation, kompilering av javaprogram	S19.1, S1.1-5, S9.4
	fre	7	Testning, enhetstestning med JUnit	<i>mtrl undelas</i>
37	mån	8	Arv	S10.1-3, S10.5
	tis	9	Polymorfism	S10.4, S10.6
	fre	10	Abstrakta klasser och gränssnitt	S10.7-8
38	mån	11	Grafiska användargränssnitt	S6, S12.1-9, S14
	tis	12	Grafiska användargränssnitt forts.	S8
	fre	13	MVC-modellen, observer-mönstret	<i>mtrl undelas</i> , S12.10
39	mån	14	Strukturerad felhantering	S11
	tis	15	Filer och strömmar	S16
	fre	16	Aktiva objekt och trädar	S13
40	mån	17	Kopiering av objekt, serialisering, objektströmmar	<i>mtrl undelas</i> , S10.10.4, S16.4.2
	tis	18	Objektrelationer. Likhet, ordningar och hashkoder.	S10.10.1-3, S10.11
	fre	19	Applikationsdesign, utvecklingsprocess, modellering	S4
41	mån		<i>reserv</i>	
	tis	20	Designmönster, Singleton, Abstract factory m.fl.	<i>mtrl undelas</i> , S sid. 186, 341
	fre		<i>reserv</i>	
42			<b>Tentamen: tor 20/10 2011</b>	

## Demo

Object oriented programming, DAT042, D2, 11/12, lp 1

Lecture 1 10

## Fundamental concepts

- object
- class
- instance variable (field)
- method
- parameter
- data type

Object oriented programming, DAT042, D2, 11/12, lp 1

Lecture 1 11

## Objects and classes

- objects
  - represent 'things' from the real world, or from some problem domain (example: "the red car down there in the car park")
- classes
  - represent all objects of a kind (example: "car")

Object oriented programming, DAT042, D2, 11/12, lp 1

Lecture 1 12

## Methods and parameters

- Objects have operations which can be invoked (Java calls them *methods*).
- Methods may have parameters to pass additional information needed to execute.

Object oriented programming, DAT042, D2, 11/12, lp 1

Lecture 1 13

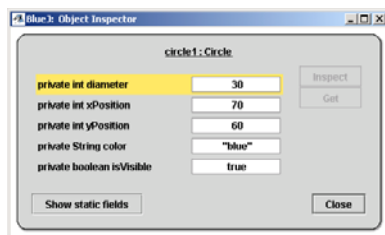
## Other observations

- Many *instances* can be created from a single class.
- An object has *attributes*: values stored in *instance variables*.
- The class defines what instance variables an object has, but each object stores its own set of values (the *state* of the object).

Object oriented programming, DAT042, D2, 11/12, lp 1

Lecture 1 14

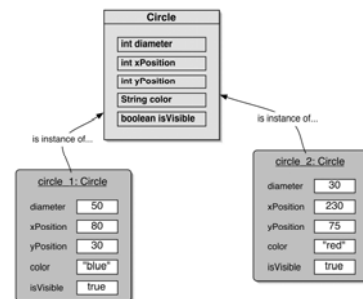
## State



Object oriented programming, DAT042, D2, 11/12, lp 1

Lecture 1 15

## Two circle objects



Object oriented programming, DAT042, D2, 11/12, lp 1

Lecture 1 16

## Source code

- Each class has source code (Java code) associated with it that defines its details (fields and methods).

Object oriented programming, DAT042, D2, 11/12, lp 1

Lecture 1 17

## Return values

- Methods may return a result via a return value.

Object oriented programming, DAT042, D2, 11/12, lp 1

Lecture 1 18