

Vergelijkende Studie van Imperatieve Programmeertalen

- Eric Steegmans
- 1e semester
- 6 studiepunten

Study Programs

- ◆ 3e Bachelor Informatica
 - ◆ Informaticakeuzeopleidingsonderdeel in alle minoren uitgezonderd de technische minor
- ◆ 1e en 2e Master Informatica
 - ◆ Verdiepend pakket software engineering
 - ◆ Verdiepend pakket veilige software
- ◆ 2e Master Computerwetenschappen
 - ◆ Kernopleiding

Objectives

- ◆ Study the most **specific concepts** offered by the current generation of object-oriented programming languages
 - ◆ Concepts for developing **specifications and implementations** of classes of objects
 - ◆ Concepts for **deriving class definitions** from already existing classes (inheritance)
 - ◆ Miscellaneous topics such as **meta-classes, generic classes and exceptions**
- ◆ Aspects related to **older generations** of programming languages are ignored as much as possible
 - ◆ Concepts related to the **implementation of methods** (loop statements, recursion, expressions, ...)
 - ◆ **Primitive types** and related operations

Programming Languages

- ◆ The course focuses on a **representative set** of object-oriented programming languages.
 - ◆ This year's set includes the programming languages C++, Eiffel, Smalltalk, Java and C#
 - » Concepts offered by Java serve as a reference set
- ◆ Next to a pure study of the concepts offered by each of these languages, an **evaluation** is forwarded as well
 - ◆ Object-oriented programming languages are evaluated in view of criteria such as **efficiency**, **simplicity** and **robustness**
- ◆ A number of **heuristics** are formulated setting out a proper and common style in each of the languages

Course Material

- ◆ A **course text** is available discussing and evaluating concepts in C++, Eiffel, Smalltalk, Java and C#
 - ◆ The text is distributed in **paper form** by the student organizations VTK and WINA
 - » The text is not completely finished
 - » The final version (on paper) will be distributed at the end of the semester
 - ◆ The preliminary text is offered electronically in PDF-format on the homepage (Toledo)
- ◆ Copies of **slides** (2 per page) are also available electronically
 - ◆ The slides posted on the homepage were used to present the material of the course in **more traditional colleges**
 - » The slides offer an alternative for studying the material of the course
 - » The slides are not completely up-to-date with the text

Programming Environments

- ◆ **Programming environments** are offered for each of the languages C++, Eiffel, Smalltalk and C#
 - ◆ The home page of the course offers [links](#) to the following compilers:
 - » GNU compiler for C++
 - Some IDE's for Windows platforms are also offered
 - » SmartEiffel compiler for Eiffel (SmartEiffel)
 - An Eclipse plugin and the original Eiffel Studio is also offered
 - » GNU compiler for Smalltalk
 - Some IDE's for different platforms are offered as well
 - » SDK for C# that only runs on Windows platforms
 - Mono is an alternative for other platforms (Linux, Mac's, ...)
 - ◆ All the compilers (environments) can be [downloaded for free](#)
- ◆ Compilers and environments for each of the languages are also **installed in the lab** of the department

- ◆ Concepts of other programming languages are studied by **translating pieces of Java code**
 - ◆ Relevant concepts are **briefly summarized** at the beginning of an exercise
 - » Students should have studied those concepts in advance
 - ◆ Translations of relevant pieces of Java code are **worked out interactively**
 - » The intention of the course is not to work out a straightforward translation
 - » The Java code merely serves to introduce the aspects to be included
- ◆ **A possible solution for each session** is distributed via the homepage (Toledo) of the course
 - ◆ The **initial assignment** will be distributed as well
 - » The distributed assignment may be more extended than the one covered during the sessions
 - ◆ The **slides** used to introduce relevant concepts are distributed as well

- ◆ Theoretical sessions are complemented with **practical sessions**
 - ◆ Practical sessions serve to **learn the compilers and environments** for experimenting with the different languages
 - » Students are encouraged to experiment as much as possible on their own machine or in the lab of the department
 - ◆ **Four sessions** are organized in the lab of the department
 - » Each session covers one of the programming languages C++, Eiffel, Smalltalk and C#

Examination

- ◆ The examination for this course is **open-book**
 - ◆ Course notes, copies of the slides and solutions for exercises may be used during the examination
 - » Other books, personal computers, mobile phones, ... are NOT allowed
 - ◆ The examination is **limited in time** (4 hours)
 - » It is not the intention of an open-book exam to study the material during the examination
- ◆ The examination is **partly oral**, and **partly written**
 - ◆ The first question is typically the development of a **small hierarchy of classes** in different programming languages
 - » A definition in Java may be given as part of the question
 - ◆ The second question consists of **small, multiple-choice assignments**, often used in testing knowledge of programming languages
 - » Such assignments typically ask for the contents of a variable after executing some piece of code
 - ◆ An **example set of questions** is available on the home page