DM22 Programming Languages

Rolf Fagerberg

Spring 2007

Programming Languages

Imperative

Assignment, sequence, decision, iteration, sub-routine/ procedure/function/method, variables hold state.

Object-Oriented

Same + well-developed modularisation tools.

Functional

No assignments, no state. Only functions and expressions. Program text close to math.

Logic

Declarative, logic based. Automated deduction from a program of facts and rules in order to prove a goal true.

Special purpose

Scripting, web, macros, concurrency,. . .

Why Study Programming Languages?

- New ways to express ideas in programming.
- New perspectives on known ways.
- Raised level of abstraction.
- Choose right language for task/problem domain.
- Pick up new languages more easily.

The language determines what can be (easily) expressed

It influences how you think about programming.

Programming Language Courses

Two options:

- Compare a large number of languages for similar and different features (language phylogeny).
- Learn to program seriously in a few languages complementing your current knowledge.

DM22 uses second option.

Teaches programming in Haskell (functional) and Prolog (logic).

Related courses: DM17 (language syntax, computational power of languages), DM18 (language semantics, compilation/language implementation).

Why Study Functional Programming

High level of abstraction. Shorter and cleaner code, faster development. Fewer bugs. Easier to prove program behavior. Well suited for parallel exectution.

Many language features not found (yet) in most imperative languages (rich type systems, type inference, type polymorphism, high-order functions, pattern matching).

Standard academic language: Haskell. Some languages in industrial use: Erlang, OCAML. Trend: Multi-paradigm languages (OO + functional): Scala, Python.

Why Study Logic Programming

Declarative. Automated deduction/state space exploration. Easy structural matching of objects.

Programming formalism well suited for certain areas (expert systems, language recognition, AI, some areas of databases).

Extendible to constraint programming. Standard language (academia and industry): Prolog.

DM22 Formally

Literature

- R. Bird: *Introduction to Functional Programming using Haskell*, 2nd edition.
- W.F. Clocksin and C.S. Mellish: *Programming in Prolog*, 5th edition.

Exam

Written exam, 4 hours, 13-scale.

Compulsory Projects

One in Haskell, one in Prolog, pass/fail.

Hours

Mondays 10–12.
Thursdays 10–12.