

# Fundamental Concepts of Programming Languages

## Lecture Presentation

conf. dr. ing. Ciprian-Bogdan Chirila  
chirila@cs.upt.ro

Department of Computing and Information Technology  
University Politehnica Timisoara

October 4, 2022



- conf. dr. ing. Ciprian-Bogdan Chirila
- chirila@cs.upt.ro
- <http://www.cs.upt.ro/~chirila>
- PhD at UPT and University of Nice - Sophia-Antipolis, France;
- PhD topic Reverse-Inheritance programming language mechanism;
- related talks given at University of Nice, France; University of Bonn, Germany;



## Meeting Bjarne Stroustrup (C++) in Nice 2003



# Definition and Implementation of Programming Languages

- programming language definition;
- the place of the programming language in the software development process;
- programming languages qualities:
  - i) consistency with the usual notation;
  - ii) readability;
  - iii) exception handling;
  - iv) automatic formal checking;
  - v) orthogonality;
  - vi) uniformity;
  - vii) scalability;
  - viii) portability;
  - ix) efficiency;



# Programming Languages Paradigms

- imperative programming languages;
- functional programming languages;
- declarative programming languages;
- sequential programming;
- concurrent programming;
- parallel programming;
- distributed programming;
- client / server model;
- short history of programming languages development: Algol, Cobol, Lisp, APL, Snobol, PL/I, Algol 68, Simula 67, Pascal, ML, C, Mesa, Concurrent Pascal, CLU, Modula 2, Ada, Prolog, Common Lisp, Standard ML, Miranda, Haskell, SmallTalk, C++, Object Oberon, Eiffel, Java, C#, ...;



# Programming Languages Definitions

- Formal programming language definition;
- Syntax;
- Syntax grammars;
- Syntax diagrams;
- Semantics;
- Operational semantics;
- Attributed grammars;
- Axiomatic semantics;
- Denotational semantics;



# Programming Languages Implementations

- Implementation of programming languages;
- Interpretation;
- Translation;
- Comparisons;
- The compiling process;
- Compiler structure;
- Analysis and synthesis;
- e.g. Compiling an assignment instruction;



# Program Entities Attributes. Attributes Dynamic Binding

- Variable domain;
- Variable lifetime;
- Memory allocation;
- Variable value;
- Variable type;





# Parameter Transmission. Generic Subprograms

- Parameter transmission by reference;
- Parameter transmission by copying;
- Parameter transmission by name;
- Parameter transmissions in different programming languages;
- Transmitting subprograms as parameters;
- Generic subprograms;



# Data Types

- Predefined types;
- Programmer defined types;
- Scalar types;
  - Structured data types;
  - Cartesian product;
  - Finite projection;
  - Sequence;
  - Recurrence;
  - Variable reunions;
  - Sets;
- Pointer type;
- Type compatibility;



# Typing Systems

- Pascal Typing System;
- C, C++ Typing System;
- Java, C# Typing System;
- Ada Typing System;
- Lisp Typing System;
- Python Typing System;
- Comparisons;
- Strongly typed programming languages;



# Abstract Data Types

- type Stack
- module Stack;
- type Stack alternative;
- module Stack alternative;



# Object-Oriented Programming Languages

- Object-oriented programming;
  - Inheritance;
  - Dynamic binding;
- Object-oriented programming in Java;
  - General aspects
  - Example
- Object-oriented programming in C#;
- Object-oriented programming in Python;
- Object-oriented programming in JavaScript;
- Object-oriented programming in TypeScript;
- Object-oriented programming in Lisp;
  - Defining objects and accessing their components;
  - Slot inheritance
  - Method inheritance and message passing;



# Control Structures

- Instruction level control structures;
  - Sequence;
  - Selection;
  - Repetition;
- Subprograms;
  - Side effects;
  - Pseudonyms;
- Exception handling;
  - in Ada;
  - in C++;



# Lambda Calculus

- Lambda expressions and functions;
- $\beta$  reductions;
- Variable binding;
- Free variables and bound variables;
- Name conflicts;
- $\alpha$  conversions;
- $\eta$  reductions;
- Boolean values and conditional expressions;
- Applied  $\lambda$  calculus;
- Evaluations order;
- Church-Rosser Theorems;



# Lab Works

- C Sharp programming language;
- OOP concepts;
- work in progress...

