2IP15 Programming Methods

From Small to Large Programs

Tom Verhoeff
Eindhoven University of Technology
Department of Mathematics & Computer Science
Software Engineering & Technology Group

www.win.tue.nl/~wstomv/edu/2ip15

Overview

- Overall structure of (Java) programs
- Static view
- Dynamic (behavioral) view

Programs: Syntax and Semantics

- Syntax: What form can a program have?

  Static aspect, independent of execution

  Program text: should conform to the grammar of the programming language

  Program executable: in another language, generated by compiler

- Semantics: What is the meaning of a program

  Dynamic aspect

  Behavior, execution

Low-level (static, syntactic) Structure of a Java Program

Bottom-up view of Java program text:

- Bits, grouped into
- Unicode Characters, grouped into
- Lexical Tokens, forming
- Definitions, Declarations, Expressions, Statements, ...

High-level Static Structure of a Java Program

Program text (processed by Java compiler):

- Collection of `.java` text files (compilation units)
- Organized in a containment hierarchy of packages
- Each file defines one or more classes or interfaces
- Each class defines (possibly static) variables and methods (later: also *nested classes*)
- One designated `public static void main(String[])` method

Executable after compilation; run by Java Virtual Machine (JVM):

- Hierarchical collection of `.class` or `.jar` binary files
- One designated `main` entry method

Behavior: Different Views on State and State Change

State (of a computation):

- Memory contents
- Variables, each 'having' a value
- Abstract data

State change (also known as *state transitions*):

- Machine instructions, operating on the memory
- Statements, operating on the variables
- Actions/operations, operating on the abstract data

State of a Running Java Program

- Collection of classes, and objects instantiated from classes
- Each variable holds a value of a primitive type or a reference to an object, forming a labeled directed graph (network of objects)
- Unreachable objects can be removed by the Garbage Collector
- Stack of nested method invocations (calls) that are active

  At the bottom of the stack is the designated `main` method.
- Each active method invocation has parameters, local variables and a current instruction address, together forming a stack frame
- Instruction at current address of topmost stack frame is executed

Summary

- Overall structure of (Java) programs
- Static view
- Dynamic (behavioral) view