

# 1 Presentation name

Vala programming language and its compiler

# 2 Authors

Tomáš Bambas, Jiří Mašek

# 3 Abstract

Our presentation will concern a relatively young programming language, Vala, and its compiler. First, we will introduce Vala as a C# like language and describe its possibilities. Next, we will introduce some special things which differs Vala from other modern OOP languages, like reference counter or GObject dependency. The most time will take the description of the Vala compiler and Vala two phases compilation. Vala compiler at first translates Vala source code to the C source code with dependencies on GNOME libraries. Then the automatically created C code can be compiled by standard UNIX C compilers like GCC or LLVM.

Vala compiler is called valac and it is written in Vala too. In the beginning of the compilation, Vala.CodeContext object is initialized. This object holds everything important for compilation process, like compiler options and the parsed Vala code tree (AST). Vala also uses a tree to represent the C code that will be produced. C code is represented by CCode classes.

The process of the Vala source code translation consists of the following steps:

1. Initialize CodeContext with command-line options.
2. Add packages from command-line and others depending on the profile.
3. Add sources, Vala, Genie, Gir, VAPI, and C from command-line.
4. Parse everything.
5. Resolve symbols.
6. Run the Semantic Analyzer.
7. Run the Flow Analyzer.
8. Use the code generator to emit code.
9. Write out VAPI and GIDL files, if a library is being compiled.
10. Compile the generated C code. (user step)

These steps will be described in our presentation.