Gosling, James; **Joy, Bill**; **Steele, Guy**
The Java language specification. (English) Zbl 0865.68001
Amsterdam: Addison-Wesley. xxv, 825 p. (1996).

Java is a general-purpose object-oriented language providing concurrent behaviour. This issue, in a series of books dedicated to the language Java, presents a complete specification of this language and the main packages of its Application Programming Interface. The book contains 22 chapters.

After the first chapter, an Introduction, the next one, called Grammars, informally presents the context-free grammars and their use in the specification of the lexical and syntactical structure of Java programs.

Chapter 3, 'Lexical structure', specifies the basic lexical units of a Java program: white spaces, comments, identifiers, keywords, literals, separators, operators as well as the Unicode used as the character set.

Chapter 4, called 'Types, Values and Variables', presents the types used by the language, their set of values and different kinds of variables. Java is a strongly typed language, i.e. any expression has a type known at compile time. The types are divided into two groups: primitive types (boolean type, numeric type - integral and floating point -) and reference types (class type, interface type and array type). Also a special null type exists. The problem of variable initialization before use, is presented.

Chapter 5, 'Conversions and Promotions', deals with specific conversions from a type to another.

Chapter 6, 'Names', describes the name forms (simple and qualified), the declarations introducing names, the scope of a name.

Chapter 7, 'Packages', introduces the structure of a Java program, the main program unit, called package, which is similar to modules in Modula or packages in Ada. The members of a package are the compilation units and the subpackages.

Chapter 8, 'Classes', describes the Java's classes with their members: variables, methods, static initializers and constructors. The variables are divided into class variables and instance variables. The methods describe the code to be executed. Static initializers are parts of the code used to initialize a class. The constructors are special methods used to initialize new class instances.

Chapter 9 describes Java's interface types. An interface declaration introduces a new reference type whose members are constants and abstract methods.

Chapter 10 defines Java arrays. These are objects dynamically created and may be assigned to variables of type Object.

Chapter 11 describes Java's exceptions. Similar to other languages (PL/I, Ada) Java takes into account the behaviour of a program violating the semantic constraints imposed by the language and provides an alternative continuation of the program. The 'catch' clause of 'try' statement explicitely handles the exceptions occurring during the execution of a program.

Chapter 12 presents the activities occurring during the execution of a Java program. Such a program is stored as binary files associated to classes and interfaces contained. The binary files are linked together with other classes and interfaces and then loaded into a Java Virtual Machine and executed.

Chapter 13, 'Binary Compatibility', specifies the minimum standards for the binary compatibility guaranteed by all Java implementations.

Chapter 14 deals with blocks and statements. Most of them are based on C and C++. Unlike C and C++, Java has no 'goto' statement, but provides 'break' and 'continue' statements with extensions allowing to specify statement labels. 'throw' and 'try' statements for exceptions and synchronized statements for achieving mutual exclusion, are specific only for Java.

Chapter 15 specifes the meaning of Java expressions and the rules for their evaluation.

Chapter 16 describes how Java ensures that local variables are definitely set before use, while other variables are automatically initialized to a default value.

Chapter 17, 'Threads and Locks', describes the semantics of Java threads and locks. Each thread inde-

pendently executes Java code. The synchronization of the threads is made by monitors. The behaviour of monitors is explained in terms of locks.

Chapter 18 describes the possibilites of automatically generating documentation from special comments inserted in the source Java code.

Chapter 19 presents a LALR(1) grammar for Java. The solutions adopted to transform the expository grammar into this LALR(1) grammar are described.

Chapter 20 through 22 present a reference manual for the packages 'java.lang', 'java.util' and 'java.io' of the Java Application Programming Interface.

The book ends with an index, credits for quotations used in this book and a colophon showing how the book was created.

Java language is related to C and C++, but includes some concepts from other widespread languages (Modula, Ada). As the authors mention this language is intended to be a production language rather than a research language. Now Java is a mature language and maybe with some modifications will be available for a widespread use.

Reviewer: M.Gheorghe (Bucureşti)

MSC:

68-01 Introductory exposition (textbooks, tutorial papers, etc.) pertaining to computer science

68N15 Theory of programming languages

Cited in **106** Documents

Keywords:

Java; object-oriented language; context-free grammars