Java

Canonical name	Java
Date of creation	2013-03-22 16:52:10
Last modified on	2013-03-22 16:52:10
Owner	PrimeFan (13766)
Last modified by	PrimeFan (13766)
Numerical id	9
Author	PrimeFan (13766)
Entry type	Definition
Classification	msc 68N15
Defines	Java

Java is a software platform intended to modernize programming on a universal scale. The chief inventor was James Gosling working for Sun Microsystems which has retained the governance of the Java platform specifications. A strong community of other companies and users has developed and affected the development chiefly IBM and Apple.

Gosling's hope for development on the platform was encompassed in the slogan

> "Write once, run anywhere."

Unfortunately, the realization has often followed the mocking slogan:

> "Write once, debug everywhere."

The name Java is a reference to coffee but was not the original name desired by Gosling.

1 The platform

The platform is comprised of several levels of abstraction.

- Every Java platform operates with a core kernel which is designed and implemented for specific physical settings. There are Java kernels targeted for multiprocessor Sparc and Itanium server environments all the way down to StrongArm and even Smart-Card chips. IBM has even prototype processors equipped to operate as solid state Java kernels. Access to this layer from the other platforms is achieved through `jave.lang` packages as well as through the Java Native Interface.

  The simplest method to create a kernel is to build an interpreter in the language and tools of a given operating system, such as Microsoft Windows, Apple's OS-X, and various Linux and Unix distributions. This unnecessary layer is generally responsible for slower performance on time critical tasks and may be responsible for much of the limited adoption of platform for large scale home or office applications.

  The most notable divergence of the kernel from C style applications is garbage collection. Memory in Java is managed completely by the kernel and so programmers cannot use dangerous tricks such as incrementing or decrementing pointers beyond the bounds of the array or structure allocated to pointer. However, by retaining deletion controls, some critical optimizations may not be able release resources when desired.

- The second layer is the programming language and its suite of standard packages. The language uses the C syntax but unlike C++ makes an explicit use of object oriented structures. The concept of exceptions, experimentally introduced in C++, is given a robust form thus removing the need to check returns for errors. An error in Java reverses the execution direction of the thread "throwing" exception to the calling functions until some layer is capable of handling the error. An unexpected result of this is that some programs now use the exception mechanism as a valid exit strategy from a loop of recursion thus in effect providing for an unbounded number of types as the result of a method invocation.

  The language has single inheritance but allows for polymorphism through interface structures. This avoids the danger of C++ multiple inheritance.

  The various packages, for example: `java.util`, `java.awt`, `java.net`, etc. provide uniform access to standard features of most hardware systems. For example, data structures such as lists, trees, queue, etc. are provided in the utility package. AWT controls standard windowing controls which allows for elegant and simply built graphical user interfaces. Packages such as the net package provide very direct methods to access internet controls (TCP/IP stacks) and provide conversions between various formats on the web. This makes network applications very favorable as there is no concern over which operating systems are linked together through the programs.

- At the most abstract layer to the platform are the extended packages. These include packages to control databases, XML utilities, 3-D packages, speech and natural language, and perhaps the most comprehensive dedication to accessibility and internationalization.

2 The standards

Having developed the Java platform from the ground up there was a rare opportunity to establish conventions to form a uniform development look which would promote ease of development. Data types are named with a capital letter for each new word or initial. For example, the data structure encoding a list is written `List`. The exception to this rule are the primitive data types of integers, floating points, etc. These are written `int`, `long`, `boolean`, etc.

In contrast, variables (references) begin with a lower case but may have upper case letters for subsequent words or initials. For example, an instance of an object `LargeInteger` could be named `largeInt`.

Unlike the Microsoft Hungarian notation, references are not labeled with short hand letters revealing their type. Because Java is a strongly typed programming language, much of the type problems of C/C++ solved by Hungarian notation is unnecessary as the problems are caught at compile time. The trade off is the inflexibility of programming in that data must be explicitly cast to the appropriate data type even when provably clear in the program. Later languages, such as Python, are in sharp contrast to Java in this regard.

At larger level, the names of new packages carry with them the website names of the developers. For example `com.sun.java.swing` or `com.ibm.java.speech` are packages from Sun Microsystems and IBM, as is easily detected in the package names. Linking between libraries is thus simply a matter of identifying the package name and is not affected by separate MAKE files etc. as is common in C/C++ settings.

3 Mathematical and scientific applications

In the world of mathematical and scientific computation, the use of Java is limited. Most theoretical applications are encumbered by unnecessary user interfaces and the strong need to optimize makes garbage collects and especially interpreters very unfavorable. The one place were the Java platform is poised for use on large scale mathematical projects is in distributed computing problems such as searches for primes etc.

3.1 External links

`http://www.mirwoj.opus.chelm.pl/ca/mjcell/mjcell.html`Mirek's Java Cellebration v.1.50