Scanned PDF

Table 2.1 (*Contd*)

Year	*Development*
1994	The team developed a Web browser called "HotJava" to locate and run applet p Internet. HotJava demonstrated the power of the new language, thus making it instantly po the Internet users.
1995	Oak was renamed "Java", due to some legal snags. Java is just a name and is not Many popular companies including Netscape and Microsoft announced their support to Ja
1996	Java established itself not only as a leader for Internet programming but also as a gene object-oriented programming language. Sun releases Java Development Kit 1.0.
1997	Sun releases Java Development Kit 1.1 (JDK 1.1).
1998	Sun relases the Java 2 with version 1.2 of the Software Development Kit (SDK 1.2).
1999	Sun releases Java 2 Platform, Standard Edition (J2SE) and Enterprise Edition (J2EE)
2000	J2SE with SDK 1.3 was released
2002	J2SE with SDK 1.4 was released.
2004	J2SE with JDK 5.0 (instead of JDK 1.5) was released. This is known as J2SE 5.0.

The most striking feature of the language is that it is *a platform-neutral* language. Ja programming language that is not tied to any particular hardware or operating system. Progra in Java can be executed anywhere on any system. We can call Java as a revolutionary technol has brought in a fundamental shift in how we develop and use programs. Nothing like this ha the software industry before.

2.2 JAVA FEATURES

The inventors of Java wanted to design a language which could offer solutions to some of the problems encountered in modern programming. They wanted the language to be not only reliable, portable and distributed but also simple, compact and interactive. Sun Microsystems officially describes Java with the following attributes:

Java 2 Features	*Additional Features*
• Compiled and Interpreted	• Ease of Developu
• Platform-Independent and Portable	• Scalability and Pe
• Object-Oriented	• Monitoring and M
• Robust and Secure	• Desktop Client
• Distributed	• Core XML Supp
• Familiar, Simple and Small	• Supplementary c
• Multithreaded and Interactive	• JDBCRowSet
• High Performance	
• Dynamic and Extensible	

Although the above appears to be a list of buzzwords, they aptly describe the full language. These features have made Java the first application language of the World Wi also become the premier language for general purpose stand-alone applications.

Compiled and Interpreted

Usually a computer language is either compiled or interpreted. Java combines both thes making Java a two-stage system. First, Java compiler translates source code into what is instructions. Bytecodes are not machine instructions and therefore, in the second stag generates machine code that can be directly executed by the machine that is running the can thus say that Java is both a compiled and an interpreted language.

Platform-Independent and Portable

The most significant contribution of Java over other languages is its portability. Java programs c moved from one computer system to another, *anywhere* and *anytime*. Changes and upgrades systems, processors and system resources will not force any changes in Java programs. This i why Java has become a popular language for programming on Internet which interconnects di of systems worldwide. We can download a Java applet from a remote computer onto our loca Internet and execute it locally. This makes the Internet an extension of the user's basic syste practically unlimited number of accessible applets and applications.

Java ensures portability in two ways. First, Java compiler generates bytecode instruction implemented on any machine. Secondly, the size of the primitive data types are machine-indep

Object-Oriented

Java is a true object-oriented language. Almost everything in Java is an *object*. All program c reside within objects and classes. Java comes with an extensive set of *classes*, arranged in *p* we can use in our programs by inheritance. The object model in Java is simple and easy to exte

Robust and Secure

Java is a robust language. It provides many safeguards to ensure reliable code. It has strict com run time checking for data types. It is designed as a garbage-collected language relieving the virtually all memory management problems. Java also incorporates the concept of excep which captures series errors and eliminates any risk of crashing the system.

Security becomes an important issue for a language that is used for programming on In of viruses and abuse of resources are everywhere. Java systems not only verify all memo also ensure that no viruses are communicated with an applet. The absence of pointers in Jav programs cannot gain access to memory locations without proper authorization.

Distributed

Java is designed as a distributed language for creating applications on networks. It has the a both data and programs. Java applications can open and access remote objects on Internet as can do in a local system. This enables multiple programmers at multiple remote locations to c work together on a single project.

Simple, Small and Familiar

Java is a small and simple language. Many features of C and C++ that are either redundan unreliable code are not part of Java. For example, Java does not use pointers, preprocesso goto statement and many others. It also eliminates operator overloading and multiple inherita detailed comparison of Java with C and C++, refer to Section 2.3.

Familiarity is another striking feature of Java. To make the language look familiar programmers, it was modelled on C and C++ languages. Java uses many constructs of C therefore, Java code "looks like a C++" code. In fact, Java is a simplified version of C++.

Multithreaded and Interactive

Multithreaded means handling multiple tasks simultaneously. Java supports multithrea his means that we need not wait for the application to finish one task before beginnin example, we can listen to an audio clip while scrolling a page and at the same time download an applet from a distant computer. This feature greatly improves the interactive performance of graphical applications.

The Java runtime comes with tools that support multiprocess synchronization and construct smoothly running interactive systems.

High Performance

Java performance is impressive for an interpreted language, mainly due to the use of intermediate bytecode. According to Sun, Java speed is comparable to the native C/C++. Java architecture is also designed to reduce overheads during runtime. Further, the incorporation of multireading enhances the overall execution speed of Java programs.

Dynamic and Extensible

Java is a dynamic language. Java is capable of dynamically linking in new class libraries, methods, and objects. Java can also determine the type of class through a query, making it possible to either dynamically link or abort the program, depending on the response.

Java programs support functions written in other languages such as C and C++. These functions are known as *native methods*. This facility enables the programmers to use the efficient functions available in these languages. Native methods are linked dynamically at runtime.

Ease of Development

Java 2 Standard Edition (J2SE) 5.0 supports features, such as Generics, Enhanced for Loop, Autoboxing or unboxing, Typesafe Enums, Varargs, Static import and Annotation. These features reduce the work of the programmer by shifting the responsibility of creating the reusable code to the compiler. The resulting source code is free from bugs because the errors made by the compiler are less when compared to those made by programmers. Thus, each of the linguistic features is designed to develop Java programs in an easier way.

Scalability and Performance

J2SE 5.0 assures a significant increase in scalability and performance by improving the startup time and reducing the amount of memory used in Java 2 runtime environment. For example, the introduction of the class, data sharing in the Hotspot Java Virtual Machine (JVM) improves the startup time by loading the core classes from the jar files into a shared archive. Memory utilization is reduced by sharing data in the shared archive among multiple JVM processes. In the earlier versions, the data was replicated in each JVM instance.

Monitoring and Manageability

Java supports a number of APIs, such as JVM Monitoring and Management API, Sun Management Platform Extension, Logging, Monitoring and Management Interface, and Java Management Extension (JMX) to monitor and manage Java applications. For example, Java provides JVM Monitoring and Management API to track the information at the application level and JVM level when deploying a large application. Java provides tools, such as jconsole, jps, jstat, and jstatd to make use of monitoring and management facilities. For example, GUI based tool called jconsole is used to monitor the JVM.

Desktop Client

J2SE 5.0 provides enhanced features to meet the requirements and challenges of the Java desktop users. It provides an improved Swing look and feel called Ocean. This feature is mainly used for developing graphics applications that require OpenGL hardware acceleration.

Miscellaneous Features

In addition to the above features, J2SE 5.0 supports the features such as:

Core XML Support J2SE 5.0 adds a powerful XML feature to the Java platform. Java contains some special packages for interface, to instantiate Simple API for XML (SAX) and Document Object Model (DOM) parsers to parse an XML document, transform the content of an XML document, and validate an XML document against the schema.

Supplementary Character Support Java adds the 32-bit supplementary character support as part of the Unicode 4.0 support. The supplementary characters are encoded with UTF-16 values to generate a different character called, *surrogate codepoint*.

JDBC RowSet Java supports JDBC RowSet to send data in a tabular format between the remote components of a distributed enterprise application. JDBC RowSet contains CachedRowSet and WebRowSet objects. The CachedRowSet object is a JavaBean component which acts like a container. This object contains a number of rows of data, which are retrieved from the database. The data stored in the CachedRowSet can be directly accessed without connecting to the database or any other data source. The rows of data that are retrieved from the database can be synchronized later. The WebRowSet object can operate without being connected to the database or data source. The WebRowSet object uses XML format to read and write the rowset.

2.3 HOW JAVA DIFFERS FROM C AND C++

Although Java was modelled after C and C++ languages, it differs from C and C++ in many ways. Java does not incorporate a number of features available in C and C++. For the benefit of C and C++ programmers, we point out here a few major differences between C/C++ and Java languages.

Java and C

Java is a lot like C but the major difference between Java and C is that Java is an object-oriented language and has mechanism to define classes and objects. In an effort to build a simple and safe language, the Java team did not include some of the C features in Java.

- *Java does not include the C unique statement keywords* **sizeof**, *and* **typedef.**
- *Java does not contain the data types* **struct** *and* **union.**
- *Java does not define the type modifiers keywords* **auto, extern, register, signed,** *and* **unsigned.**
- *Java does not support an explicit pointer type.*
- *Java does not have a preprocessor and therefore we cannot use* **# define, # include,** *and* **# ifdef** *statements.*
- *Java requires that the functions with no arguments must be declared with empty parenthesis and not with the* **void** *keyword as done in C.*