



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики
Кафедра системного програмування і спеціальних комп’ютерних систем

Реферат

з дисципліни «Англійська мова професійного спрямування» на тему:
«Java the programming language»

Виконав:
студент III-го курсу
групи KB-41
Горпинич-Радуженко Іван

Київ 2017

Java the programming language

Java, the programming language, is an object-oriented language targeting end-users for development of application software. The first release of Java was presented in 1995. Java was initially owned by Sun Microsystems, which was in turn bought by Oracle Corporation in 2009 for 7.4 billion dollars. In a press release from Oracle it states that "Java is one of the computer industry's best-known brands and most widely deployed technologies, and it is the most important software Oracle has ever acquired" The reason for Oracle's investment is also because their own fast growing business, Oracle Fusion Middleware, is built on Java technology meaning they now have full control over Java's further development.

Java goes under the lead words "write once, run anywhere" which means that the source code you write is platform independent and can be executed on all types of computers if they support the Java virtual machine, JVM .

The language is object oriented from the ground up, meaning that even primitive types for instance int, boolean and char have their corresponding wrapper classes Integer, Boolean and Character. Java's standard library is the java.lang package which contains all classes that are fundamental to the Java design. Java is strictly hierarchical which means that all classes build on the roots class hierarchy. For example the Object class is the root class in the java.lang package and all the other classes in the lang package inherit the methods from Object, even Integer, Boolean and Character .

Purpose of Java

The purpose of Java is explained in the so called white paper written by James Gosling and Henry McGilton in 1996 where they claim that "The design requirements of the Java™ programming language are driven by the nature of the computing environments in which software must be deployed."

The design goals of Java are to be simple, familiar, object oriented, multithreaded and interpreted. It aims to be secure and reliable since the purpose of Java is to operate in distributed environments.

Gosling and McGilton state that "the fundamental concepts of Java technology are grasped quickly; programmers can be productive from the very beginning".

Java is said to be familiar because it has inherited a C++ look (C++ was the language of tongue at this time) so the idea of adopting a C++ syntax was to attract C++ programmers to try out Java. Java is said to be more secure than other languages because its compiler catches a lot of errors at compile time, in addition the Java runtime environment catches errors at runtime. This means that it will prevent many program crashes caused by bad or forgotten error handling. The lack of pointer arithmetic in the language reduces the risk of pointing to wrong or non-existing addresses that can also result in program crashes.

Gosling claims that "language features guide programmers towards reliable programming habits".

The removal of manual memory management "eliminates entire classes of programming errors that bedevil C and C++ programmers".

According to the design principles, Java is said to be simple since it does not need "extensive programmer training". The designers also say that "the fundamental concepts are grasped quickly" so that programmers can be productive quickly. In the white paper written by Gosling, there is a chapter about the reasons for the decisions to leave out some of the things in C and C++. They claim that in C and C++ there is a lot of context you need to understand to be able to grasp another persons code. They also claim that the usage of these C features results in a "new programming language that is incomprehensible to anybody other than its creator, thus defeating the goals of good programming practices". An example of this is the related header files and the #define. Instead of using the C++ header files, in Java, you declare other classes and methods directly in the Java source file with the import declaration. The usage of classes removes the need for structures and unions like those in C and C++.

Threads are used as a way of scheduling the computer's CPU into different processes. When using threads, one has to be careful in handling concurrency in the computers memory and registers meaning that threads have to deal with data in an ordered way. This can lead to 'hard-to-find' bugs. Threading can be done with C and C++ by accessing the native threading API. What was new with Java was that the threading usage was abstracted into the core language with the class Thread available in the standard library.

Java is both a compiled and interpreted language. This means that Java source code is compiled into Java byte code which then is interpreted by the Java Virtual Machine. It has the same features as other compiled languages but the source code becomes portable since its compiled byte code can be interpreted by the JVM.

The story behind Java and why it became so popular

In 1991 James Gosling, Mike Sheridan, Patrick Naughton and ten other programmers started working on a secret research project separated from the rest of the Sun Corporation.

The group worked under the name: Green project and the idea was to foresee the next big thing in technology and hence be the first to deploy it. The focus fell on designing a system that would combine and control all new types of home electronics, an interactive remote control. This was a new market that was different from the traditional computer business, where big money was spent on the most powerful and accurate engineered components.

The Green project group needed to establish some important key concepts that were different from what the traditional work station programmer were accustomed with. For example, in the consumer electronics market "consumers demanded lowcost, bug-free and relatively simple, easy-to-use products.". The project group also stated that the product needed to be backwards compatible since components and hardware would change with time.

To meet these requirements the team established that they needed a platformindependent process for development. Gosling, who had previously created a

pseudocode virtual machine for Pascal programs during his school years in the 80's, was not late in seeing the need of a new language that could be supported on all types of software and executed on all types of hardware.

The group started by extending the C++ compiler. After a while, this extension did not meet the requirements and the idea of a new programming language started to take form. "All along, the language was a tool, not the end," Gosling says. "This was nice in a number of ways. The goal was never 'Let's take on C++,' [but] to build a system that would let us do a large, distributed, heterogeneous network of consumer electronic devices all talking to each other.". "In 18 months, we did the equivalent of what 75-people organizations at Sun took three years to do" says Naughton.

The launch of the interactive remote control did not go as planned but the management at Sun had seen the potential of the programming language, at the time called Oak, and looked around to see if it could be useful elsewhere. Their eyes fell on interactive TV and the Green project was dissolved and restructured into another group called FirstPerson, but the market for TV business was premature and no one saw the potential in TV on demand.

At the same time, the Internet grew popular and new types of interactive web browsers were constructed, such as Mosaic later reconstructed into Netscape. The team thought that the growing web could be a potential bearer of the platform independent language. "We released the source code over the Internet in the belief that the developers would decide for themselves,..." says Lisa Friendly, FirstPerson employee and original member of the Java Products Group.

One of the reasons why Java became one of the most popular programming languages is because of Java applets. A Java applet is a small program that can be embedded in web pages. The applets enable programmers to make a more interactive web page with graphical and animated little programs. The applets enhanced the Java language claim on being safe. The applet is downloaded from a server and its code executed in a 'sand-box' environment on the browser giving limited access to the client's file systems and local storage.

Summary

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2016, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

Literature:

1. The Java Saga, [article] David Bank, Wired magazine, Issue 3.12, published [Dec 1995], seen at [2012-06-01], available at <http://www.wired.com/wired/archive/3.12/java.saga.html>
2. Oracle Buys Sun, [webpage] Oracle Press Release [2009-04-20], seen at [2012-06-01], available at <http://www.oracle.com/us/corporate/press/018363>
3. write once run anywhere, [webpage], seen at [2012-08-27], available at http://en.wikipedia.org/wiki/Write_once,_run_anywhere
4. API specification for version 6 of the Java™ Platform, [webpage], Standard Edition, seen at [2012-06-01], available at <http://docs.oracle.com/javase/6/docs/api/>
5. A White Paper, James Gosling, Henry McGiltonseen, [webpage] published [May 1996], seen at [2012-06-01], available at <http://java.sun.com/docs/white/langenv/>
6. A White Paper, James Gosling, Henry McGiltonseen, [webpage] published [May 1996], seen at [2012-06-01], available at <http://java.sun.com/docs/white/langenv/Intro.doc2.html>
7. Wikipedia about threads in computers, [webpage], seen at [2012-06-01], available at [http://en.wikipedia.org/wiki/Thread_\(computing\)](http://en.wikipedia.org/wiki/Thread_(computing))
8. Java: The inside story, Michael O'Connellseen, [webpage] published [July 1995], seen at [2012-06-01], available at <http://sunsite.uakom.sk/sunworldonline/swol-07-1995/swol-07-java.html>

Dictionary:

Deployed – розгорнутий.

Corresponding wrapper - відповідна обгортка.

Strictly hierarchical - строго ієрархічна.

Design goals - мета розробки.

Multithreaded – багатопотоковий.

Inherited – успадкований.

Memory management - управління пам'яттю.

Import – імпортувати.

Scheduling – планування.

Establish – встановити.

Compiler – компілятор.

Elsewhere - в іншому місці.

Equivalent – еквівалент.