Um pouco sobre a linguagem de Java

● Surgimento

A linguagem Java começou a surgir em 1991 na Sun Microsystems. Inicialmente era parte de outro projeto, chamado Green Project, que tinha como objetivo possibilitar a convergência entre computador, equipamentos eletrônicos e eletrodomésticos. (LUCKOW; MELO, 2010, p. 25).

A tecnologia Java foi criada como uma ferramenta de programação de um projeto da Sun Microsystems, chamado The Green Project, iniciado por Patrick Naughton, Mike Sheridan e James Gosling, em 1991. Esse projeto tinha como principal objetivo criar uma nova plataforma para a computação interativa, ou seja, a linguagem de programação não era o principal objetivo do projeto. No verão de 1992 foi gerada a primeira demonstração do projeto, que representou um sistema executando em um handheld com capacidade de controle remoto que ainda oferecia uma interface sensível ao toque (touchscreen) interativa. Esse handheld foi chamado de \*7 (star seven), por esta ser a forma de atender (ou puxar) chamadas telefônicas entre os telefones dos integrantes da equipe. (MENDES, 2009, p. 16).

● Características

A linguagem de programação Java representa uma linguagem simples, orientada a objetos, multithread, interpretada, neutra de arquitetura, portável, robusta, segura e que oferece alto desempenho. É importante observar que a tecnologia Java é composta de uma linguagem de programação e de uma plataforma (API e a máquina virtual). (MENDES, 2009, p. 17).

Uma das características marcantes dessa linguagem é a sua portabilidade. Um programa portável é aquele que necessita de pouca ou nenhuma alteração no seu código para poder ser executado em plataformas diferentes. As plataformas, compostas basicamente por hardware e sistema operacional, podem ser bastante diferentes entre si, o que impossibilita, na maioria das vezes, a execução de um mesmo programa em plataformas distintas. Para oferecer portabilidade, Java utiliza processos de compilação e interpretação para execução de programas. Um programa para ser executado deve ser compilado pelo compilador Java e posteriormente interpretado. (COSTA, 2008, pg. 1).

Definições

● JVM

Uma máquina virtual Java é um representante, um garoto de recados, um intermediário. A JVM age como um intérprete entre o bytecode Java executável em qualquer lugar e o sistema do seu computador. Enquanto trabalha, a JVM conduz seu computador durante a execução das instruções descritas nele. Ela interpreta o bytecode para o seu sistema Windows, Mac ou Linux ou qualquer outro computador. Isto é bom e é o que torna os programas Java mais portáveis do que outros em qualquer outra linguagem. (BURD, 2013, pg. 33).

“Um programa JVM recebe bytecode como entrada e interpreta as instruções como se fosse um processador físico executando código de máquina” (tradução livre). (LINDSEY; TOLLIVER; LINDBLAD, 2005).

● JRE

A versão do Java que um usuário comum possui instalada em sua máquina é chamada de Java Runtime Environment (JRE). Isso possibilita a execução de programas Java, e programas Java não serão executados em ambientes sem o JRE instalado. (LASSOFF, 2017, pg. 10, tradução nossa).

O JRE é uma coleção de programas que contém a JVM e muitas bibliotecas/arquivos de classe que são necessários para a execução de programas na JVM (via comando java). (ALANKUS *et al.*, 2019, pg. 3, tradução nossa).

● JDK

“JDK consiste de várias ferramentas que são usadas para desenvolver e executar programas Java. Algumas das ferramentas incluídas no JDK são javac, java, javadoc, javap, etc.”(tradução livre). (KHURANA, 2014).

“JDK é uma coleção de ferramentas de software e bibliotecas de suporte que permitem a criação e execução de programas de linguagem Java” (tradução livre). (SAMOYLOV, 2018).

● Como eles interagem

O compilador e o interpretador Java são fornecidos gratuitamente. Há duas formas principais de distribuição desses elementos: o JDK e o JRE. O JDK (Java Development Kit) contém o compilador, a JVM, a API Java (pacotes utilitários), programas de suporte (como o jar), exemplos e documentação. O JDK é destinado ao desenvolvimento e à execução de programas. O JRE (Java Runtime Environment) é destinado à execução de programas, devendo ser utilizado quando não for necessária a criação e/ou alteração de programas escritos nessa linguagem. (COSTA, 2008, pg. 3).

Cada sistema operacional moderno vem com uma JVM específica do sistema operacional que é responsável pela tradução do bytecode para a linguagem de código de máquina correspondente naquela plataforma. A JVM faz parte de um grande sistema, o Java Runtime Environment (JRE). JDK contém ferramentas de desenvolvimento e o JRE. O JRE vem com um conjunto de classes com APIs (Application Programming Interface) Java. Programas Java necessitam da presença do JRE para executar programas Java (na forma de bytecode) naquela plataforma. (JANA, 2005, pg. 31, tradução nossa).

Tabela 1 - Tipos primitivos de dados e sua capacidade de armazenamento

| Nome do tipo | O que armazena | Capacidade |
| --- | --- | --- |
| char | Texto | valor máximo: \uffff (ou 65.535) |
| float | Valor inteiro | valor máximo: infinito positivo |
| double | Valor real | valor máximo: infinito positivo |
| int | Valor inteiro | Antes do Java SE 8:  valor máximo: 2^31-1  Java SE 8 e diante:  valor máximo: 2^32-1 |
| long | Valor inteiro | Antes do Java SE 8:  valor máximo: 2^63-1  Java SE 8 e diante:  valor máximo: 2^64-1 |
| boolean | Texto | True |

Fonte: Oracle (2022)

O que é uma IDE

O segundo tipo de ambiente de desenvolvimento é um IDE (Integrated development environment, ambiente de desenvolvimento integrado) disponível em muitas fontes como Borland’s JBuilder, Oracle’s JDeveloper e Sun’s Forte. Alguns desses são gratuitos para uso pessoal em um ambiente de não-produção. Esses IDEs são ambientes controlados por janelas e têm todos os recursos SDK além de recursos como geração de código default, depuração avançada e “tutores” automáticos de programação (code coaches). A compilação ou a execução de um programa Java pode ser feita clicando em um menu ou barra de ferramentas no IDE. (SIKORA, 2003, pg. 5).

“Um integrated development environment (IDE) é uma ferramenta tudo em um para escrever, editar, compilar e executar programas de computador”(tradução livre). (BURD, 2004).

● Eclipse

O Eclipse em si fornece apenas o ambiente integrado para a execução dos plug-ins e uns poucos plug-ins básicos, como editor de textos ASCII, sistema de ajuda e integração ao CVS. Para iniciar o desenvolvimento, em qualquer linguagem que seja, devem ser instalados plug-ins adicionais. (GONÇALVES, 2006, pg. 22).

Eclipse é o Java IDE mais usado. Ele possui um extenso sistema de plug-in que cresce o tempo inteiro, então não é possível listar todos os seus recursos. Seu recurso principal é para desenvolver aplicativos Java, mas plugins também nos permitem escrever o código em Ada, ABAP, C, C++, C#, COBOL, D, Fortran, Haskell, JavaScript, Julia, Lasso, Lua, NATURAL, Perl, PHP, Prolog, Python, R, Ruby, Rust, Scala, Clojure, Groovy, Scheme, e Erlang. (SAMOYLOV, 2018, pg. 73, tradução nossa).

Referências

ALANKUS, Gazihan *et al*. **Java Fundamentals**: a fast-paced and pragmatic introduction to one of the world's most popular programming languages. Birmingham: Packt, 2019.

BURD, Barry. **Eclipse For Dummies**. Indianápolis: Wiley Publishing, 2004.

BURD, Barry. **Java para leigos**. 5. ed. Rio de Janeiro: Alta Books, 2013.

COSTA, Daniel Gouveia. **Java em rede**: recursos avançados de programação. Rio de Janeiro: Brasport, 2008.

GONÇALVES, Edson. **Dominando o Eclipse:** tudo o que o desenvolvedor java precisa para criar aplicativos para desktop, da criação do aplicativo ao desenvolvimento de relatórios**.** Rio de Janeiro: Editora Ciência Moderna, 2006.

JANA, Debasish. **Java and Object-Oriented Programming Paradigm**. New Delhi: Prentice-Hall Of India, 2005.

KHURANA, Rohit. **Programming with Java**. New Delhi: Vikas, 2014.

LASSOFF, Mark. **Java Programming for Beginners**. Birmingham: Packt Publishing, 2017.

LINDSEY, Clark S.; TOLLIVER, Johnny S.; LINDBLAD, Thomas. **JavaTech**: an introduction to scientific and technical computing with java. Cambridge: Cambridge University Press, 2005.

LUCKOW, Décio Heinzelmann; MELO, Alexandre Altair de. **Programação Java para a Web**. São Paulo: Novatec Editora, 2010.

MENDES, Douglas Rocha. **Programação java**: com ênfase em orientação a objetos. São Paulo: Novatec Editora, 2009.

ORACLE. **Primitive Data Types**. Disponível em: https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html. Acesso em: 05 abr. 2022.

SIKORA, Michael. **Java**: Guia prático para programadores. Rio de Janeiro: Campus, 2003.

SAMOYLOV, Nick. **Introduction to programming**: learn to program in java with data structures, algorithms and logic. Birmingham: Packt, 2018.