LP3L3 – Linguagem de Programação III

2020.2

Aula 01
Introdução

Prof. Everton Silva
Prof. Argemiro Bevilacqua
everton.silva@ifsp.edu.br
bevilacqua.argemiro@ifsp.edu.br

Aula de Hoje

- Tecnologia Java
- Plataforma Java
- Linguagem Java
- Ambiente de desenvolvimento
- Primeiro programa
- Convenções de código
- Exercícios

Tecnologia Java

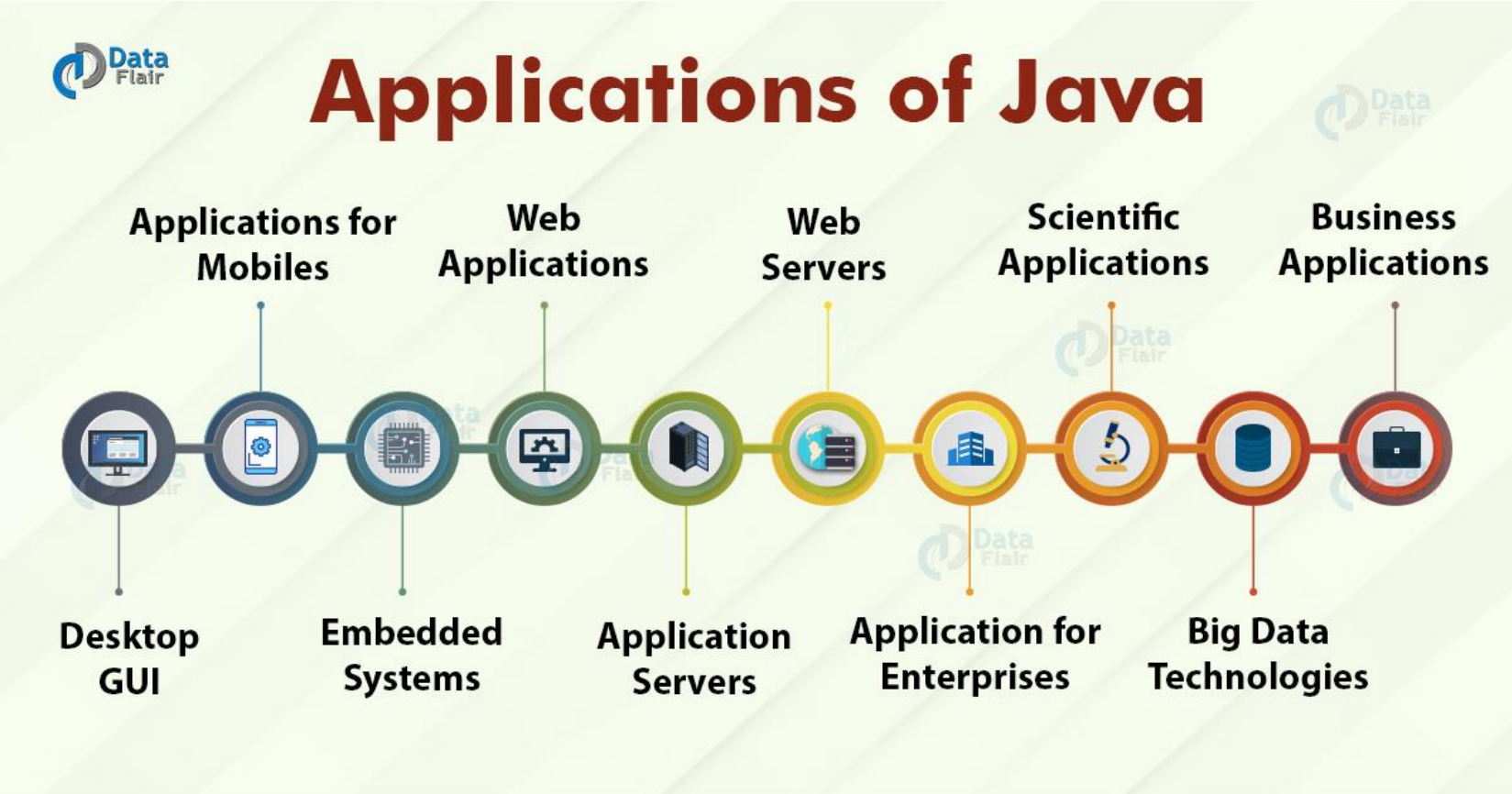
- Desenvolvida desde 1991, pela Sun Microsystems (hoje, Oracle - 2010):
  - Equipe chefiada por James Gosling – programador Canadense.

- Promessa de rodar em qualquer plataforma de hardware – independência de plataforma.

- Famoso slogan: "*Write Once*, *Run Anywhere*" (WORA).

Tecnologia Java

Alguns dispositivos que usam Java



https://data-flair.training/blogs/applications-of-java/

Tecnologia Java

Java Standard Edition – Java SE

- Contém os recursos necessários para desenvolver aplicativos de desktop e servidor;

Java Enterprise Edition – Java EE

- Desenvolver aplicativos em redes distribuída e em grande escala e também aplicativos web;

Java Micro Edition – Java ME

- Subconjunto do Java SE voltado para o desenvolvimento de aplicativos para dispositivos embarcados: *smartwatches*, MP3 players, decodificadores de TV e muitos outros;

Plataforma Java

- Normalmente, programas rodam diretamente em cima do SO, em uma máquina física.

- Programas Java executam em cima de uma máquina virtual (*JVM – Java Virtual Machine*).

- Compilador Java (comando javac)
  - Transforma código-fonte em código-objeto (arquivos .java e .class, respectivamente).

- Máquina virtual Java (comando java)
  - Executa código-objeto (arquivos .class).

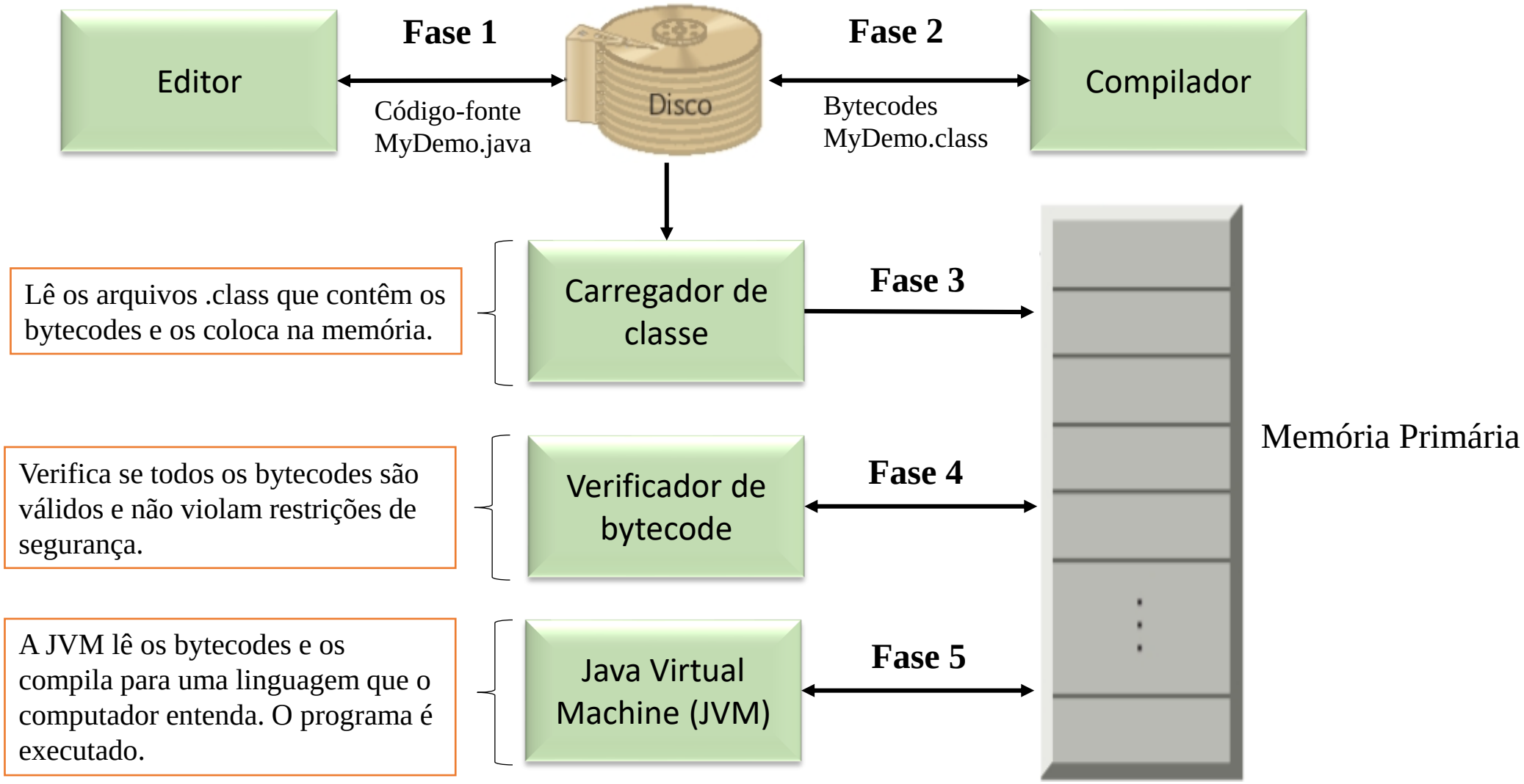
Plataforma Java

- JRE
  - Java *Runtime Environment* (Ambiente de execução).
  - Permite executar código Java.
  - Máquina virtual + bibliotecas + ferramentas.
  - Inclui o programa java.

- JDK
  - Java *Development Kit* (Kit de Desenvolvimento Java).
  - É o JRE + compilador + outras ferramentas.
  - Inclui os programas java e javac.

- Link para download:
  - http://www.oracle.com/technetwork/pt/java/javase/downloads/index.html

Ambiente de Desenvolvimento



Ambiente de Desenvolvimento

Fase 1: escrever um programa Java

- Escrever um programa Java (código-fonte) utilizando um editor e salvar na unidade de disco;
- Arquivos de código-fonte Java recebem um nome e terminam com a extensão .java;

Fase 2: compilar um programa Java em bytecodes

- Utilizar o comando javac (o compilador Java) para compilar um programa;
- Digitar o comando `javac Welcome.java` no prompt de comandos de seu SO;
- É produzido um arquivo `.class (Welcome.class)` caso o programa compile;
- A JVM executa os bytecodes pelo comando java;
- Digitar o comando `java Welcome` no prompt de comandos para executar o aplicativo;

Ambiente de Desenvolvimento

Fase 3: carregar um programa na memória

- A JVM armazena o programa na memória para executá-lo (carregamento);
- O carregador de classe pega os arquivos `.class` e transfere os bytecodes para a memória primária;

Fase 4: verificação de bytecode

- O verificador de bytecode examina e assegura que o bytecodes são válidos e não violam restrições de segurança;
- O Java certifica de que os programas Java que chegam pela rede não danifiquem os arquivos ou o sistema (como vírus ou worms do computador);

Ambiente de Desenvolvimento

Fase 5: execução

- A JVM executa os bytecodes do programa, as ações especificadas por ele;
- Os programas Java passam por duas fases de compilação:
  - o código-fonte é traduzido em bytecodes;
  - durante a execução, os bytecodes são traduzidos em linguagem de máquina para o computador real no qual o programa é executado;

Linguagem Java

- Foco em orientação a objetos.
- Sintaxe fortemente inspirada em C++.
- Sem ponteiros, com gestão automática de memória (*garbage collector*).
- Extensa biblioteca padrão de funcionalidades.

Linguagem Java

Os 5 principais objetivos da linguagem:

- Simples, orientada a objetos e familiar.
- Robusta e segura.
- Independente de arquitetura e portátil.
- Alto desempenho.
- Interpretadas - Java, C#, Python:
  - Códigos transformados em uma linguagem intermediária, posteriormente interpretada pela máquina virtual (JVM).

http://www.oracle.com/technetwork/java/intro-141325.html

Primeiro Programa

- Primeiro vamos compilar e executar o código Java sem usar uma IDE (netbeans, eclipse, bluej).

- Vamos precisar de:
  - JDK instalado e configurado.
  - Editor de texto.
  - Variável de ambiente definida no PATH para utilizar os comandos javac e java.
  - Terminal para execução de linhas de comando.

Primeiro Programa

- No Windows:
  - md projeto-ola-mundo
    cd projeto-ola-mundo
    notepad OlaMundo.java

- No Ubuntu:
  - mkdir projeto-ola-mundo
    cd projeto-ola-mundo
    gedit OlaMundo.java &

Primeiro Programa

- OlaMundo.java deve conter o seguinte código:

```
class OlaMundo {
    public static void main(String[] args) {
        System.out.println("Ola, mundo!");
    }
}
```

- Para compilar:
  - javac OlaMundo.java => gera OlaMundo.class
  - javac –d bin OlaMundo.java => gera OlaMundo.class
  - -d: informa ao compilador onde inserir o arquivo .class.

- Para executar:
  - java OlaMundo

Sintaxe Java

- Semelhante a C e C# em muitos aspectos;
- Tipos primitivos: int, char, float, double, void etc.
- Estruturas de controle e repetição: for, if, switch, while, do.
- Uso de parênteses () e chaves {}.
- Operadores: +, ++, -, --, =, ==, !=, <, <=, ? etc.
- Ponto-e-vírgula para encerrar comandos.

Sintaxe Java

- Boolean
  - C: não existe tipo booleano, usa-se int.
  - C#: bool.
  - Java: boolean.

- String
  - C: não existe tipo string, usam-se char* e char[].
  - C#: string (String também funciona).
  - Java: String.

- Vetor:
  - C: int vetor[10]; // Tamanho após nome da variável
  - C#: int[] vetor; // Tamanho após tipo
  - Java: int[] vetor; ou int vetor[]; char[] vetC;

Sintaxe Java

- Saída padrão:
  - `System.out.print(valor);`
  - Imprime sem pular linha.

  - `System.out.println(valor);`
  - Imprime pulando linha.

- Entrada padrão usando `java.util.Scanner`:

  - ```
    Scanner scanner = new Scanner(System.in);
    String strLida = scanner.nextLine();
    int intLido = scanner.nextInt();
    //O mesmo para float, double, boolean etc.
    ```
    ```

Sintaxe Java

- Escrever mais de um valor na saída:

1. Concatenação de strings
   - `System.out.println("Os valores são " + valor1 + ", " + valor2);`

2. printf
   - `System.out.printf("Os valores são %d, %d", valor1, valor2);`

Convenções de Código

- Separação de palavras alternandoCaixaAltaEBaixa (*CamelCase*)
  - `int variavel;`
  - `int variavelComMaisPalavras;`

- Classes
  - PrimeiraLetraMaiúsculaEmTodasAsPalavras
  - `String, System, InputStream`
  - `Pessoa, PessoaFisica,  PessoaJuridica`

Convenções de Código

- Objetos, variáveis em geral e métodos:
  - primeiraLetraMinúscula
  - A partir da 2ª palavra, 1ª letra maiúscula
  - `int variavelComMaisPalavras;`
  - `Pessoa pessoa = new Pessoa();`

- Métodos
  - `minhaString.toLowerCase();`
  - `pessoa.getCpf();`

- Constantes
  - TODAS_MAIÚSCULAS e com sublinhado (_) para separar palavras
  - `private static final int PORTA = 80;`

Convenções de Código

- Pacotes
  - todasminusculas
  - É a exceção à regra do CamelCase. Todas as letras são minúsculas, mesmo com várias palavras.
  - `br.edu.ifsp.cmp.programacaoemjava`
  - `meupacote.outropacote.maisum`

Exercícios

1. Escreva um programa para ler o salário de uma pessoa e imprimir o desconto do INSS segundo as regras abaixo:
   - menor ou igual a R$ 600,00 – Isento
   - maior do que R$ 600,00 e menor ou igual a R$ 1.200,00 - 20%
   - maior do que R$ 1.200,00 e menor ou igual a R$ 2.000,00 - 25%
   - maior do que R$ 2.000,00 - 30%

2. Escreva um programa que leia um vetor de caracteres (`char[]`) da entrada padrão e converta-a numa sequência de dígitos segundo um teclado de celular.

Dúvidas?


QUEM NÃO ENTENDEU, LEVANTE A MÃO
O QUE VOCÊ NÃO ENTENDEU, MANOLITO?
NADA, DESDE MARÇO ATÉ AGORA
QUINO