



Aula 1
Introdução
ao Java

Antes de Começar

- **Contraste de paradigmas de programação**
 - **Programação procedural/imperativa:** os programadores abstraem o programa como uma sequência de funções/procedimentos executadas de modo sequencialmente ou de modo empilhado e as execuções consistem na troca de estados de variáveis
 - **Programação orientada a objetos:** os programadores abstraem um programa como uma coleção de objetos capazes de realizar ações e interagirem entre si
 - A POO é um dos 4 principais paradigmas de programação (as outras são programação imperativa, funcional e lógica)

Antes de Começar

● Programação Orientada a Objetos

- Modelo de programação onde diversas classes possuem características que definem objetos na **vida real**
- Cada classe determina o comportamento do objeto definido por **métodos**, seus estados possíveis definidos por **atributos** e como os objetos interagem entre si
- Foi criado com o intuito de aproximar o mundo real do mundo virtual
- É tida como a melhor estratégia para se eliminar o “gap semântico”, dificuldade recorrente no processo de modelar o mundo real do domínio do problema em um conjunto de componentes de software

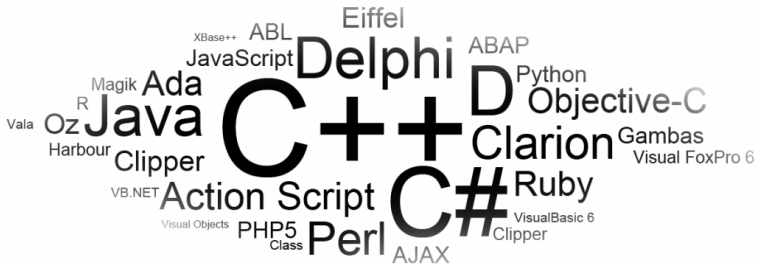
Antes de Começar

● **Vantagens da Programação Orientada a Objetos**

- Codificação mais próxima do cenário real do problema a ser resolvido
- Maior facilidade no reúso de código
- Maior facilidade na manutenção de código
- Maior rapidez na programação
- Sistema mais “interpretável”
- Maior segurança

Antes de Começar

- Existem várias linguagens orientadas a objetos























- Lista de linguagens de programação orientadas a objetos:
https://en.wikipedia.org/wiki/List_of_object-oriented_programming_languages

Antes de Começar

- Linguagens mais utilizadas e dispositivos em que podem ser utilizadas segundo a IEEE (2014) [Canaltech, 2014]:

Language Types



Language Rank	Types	Spectrum Ranking
1. Java	  	100.0
2. C	  	99.2
3. C++	  	95.5
4. Python	 	93.4
5. C#	  	92.2
6. PHP		84.6
7. Javascript	 	84.3
8. Ruby		78.6
9. R		74.0
10. MATLAB		72.6

Antes de Começar

- Dados a grande **utilização no mercado**, a **remuneração**, a **didática para o ensino de POO**, e a **semelhança com outras linguagens de programação**, nesta disciplina a linguagem de programação que será adotada é **JAVA**



Antes de Começar

- Dado também que Python e JavaScript aparecem constantemente nas listas de “linguagens mais usadas” ou “linguagens para aprender”, também serão apresentados os conceitos de orientação a objetos nessas duas linguagens



Sobre o Java

- Surgiu no início da **década de 90** na empresa **Sun Microsystems** – nos dias atuais é desenvolvida pela **Oracle**
- Java é uma linguagem **bastante didática** e é a **base de outras linguagens de programação orientadas a objetos**
- Java tornou-se a **linguagem preferida** para implementar **aplicativos baseados na Internet** e **softwares para dispositivos que se comunicam por uma rede**
- Equipamentos de som estéreo e outros dispositivos domésticos muitas vezes são conectados em rede pela tecnologia Java

Sobre o Java

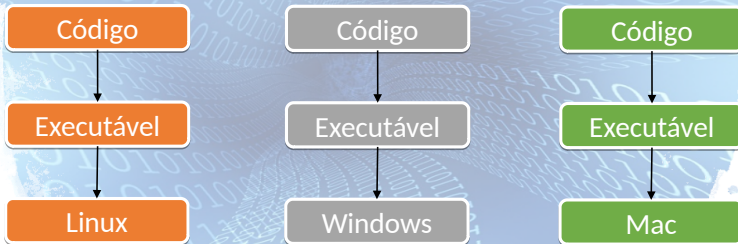
- A **sintaxe similar a outras linguagens** (C, C++, C#, ...)
- Java é a linguagem **base para o desenvolvimento de aplicações mobile para Android**, simplesmente a plataforma mobile mais utilizada no mundo
- Java é a **linguagem preferida** para atender as **necessidade de programação de muitas organizações**

Sobre o Java

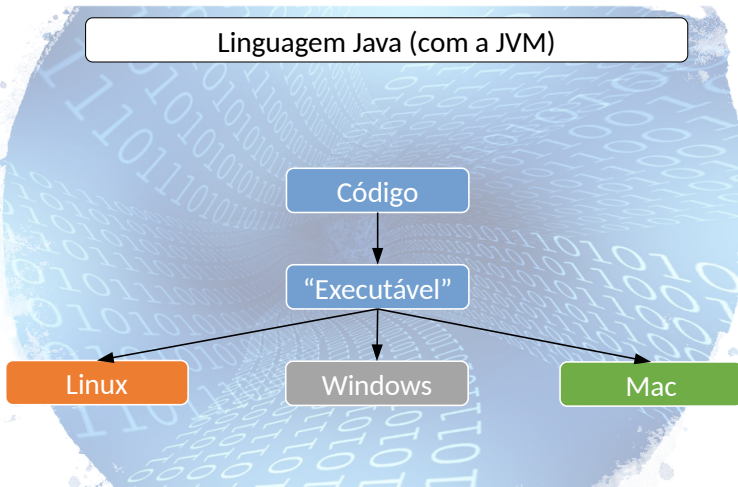
- A **compilação do código fonte** cria um “**executável**” que, por sua vez, **será interpretado por uma máquina virtual**
- A **MÁQUINA VIRTUAL JAVA** funciona como um **intermediário entre o código e a plataforma na qual esse código será executado**
- Isso permite que a aplicação seja executada em diferentes plataformas → **filosofia WORA**: *write once, run anywhere* (em português, “escreva uma vez, execute em qualquer lugar”)
- Reduz significativamente o tempo e o custo de desenvolvimento de sistemas

Sobre o Java

Linguagens compiladas (como o C)



Sobre o Java



Sobre o Java

- **O que é preciso para começar a programar em JAVA?**
 - Um computador
 - Kit para compilar e executar programas escritos em JAVA
 - **Oracle JDK:** Java SE Development Kit
(<http://www.oracle.com/technetwork/pt/java/javase/downloads/jdk8-downloads-2133151.html>)
 - **Open JDK:** versão open source do Oracle JDK
(<http://openjdk.java.net/>)
 - **OBSERVAÇÃO:** lembrar de arrumar o *classpath* no Windows

Ambiente típico de desenvolvimento JAVA

Fase 1: Edição

Editor



O programa é criado em um **editor** e **armazenado em disco** em um arquivo cujo nome termina com **.java**

Fase 2: Compilação

Compilador



O compilador cria **bytecodes** e os armazena em um arquivo cujo nome termina com **.class**

Fase 3: Carregamento

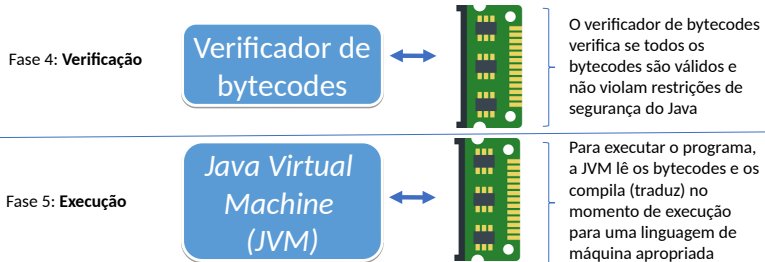
Carregador de Classes



O carregador de classe lê os arquivos **.class** que contém os bytecodes a partir do disco e coloca esses **bytecodes na memória**



Ambiente típico de desenvolvimento JAVA



Fase 1: Criando um Programa

- Utilizar um editor para gerar um código fonte
 - **Windows:** Notepad ou Notepad++
 - **Linux:** gedit, vi, emacs, pico, ...
 - **Integrated Development Environment (IDE):** Netbeans, Eclipse, BlueJ, JCreator...
 - **Online:** CodingGround
(<http://www.tutorialspoint.com/codingground.htm>)

Frase 2: Compilando um Programa Java em Bytecodes

- Utilizar um comando para compilar o(s) arquivo(s) .java

Compilando um arquivo `welcome.java` via linha de comando

```
javac welcome.java
```

- Se o programa compilar, o compilador produz um arquivo `.class` (no caso do exemplo `welcome.class`) que contém a versão “compilada” do programa

Frase 2: Compilando um Programa Java em Bytecodes

- O compilador java converte o código-fonte em **bytecodes** → instruções a serem executadas na fase de execução (Fase 5)
- Os bytecodes são executados pela **Java Virtual Machine (JVM)** → aplicativo de *software* que simula um computador, mas oculta o sistema operacional e o *hardware* subjacentes dos programas que interagem com ela

Fase 2: Compilando um Programa Java em Bytecodes

- Se a mesma VM for instalada nas várias plataformas de computador, os aplicativos que ela executa podem ser utilizados em todas essas plataformas
- Ao contrário da linguagem de máquina, que é dependente do hardware específico de computador, os bytecodes do Java são **portáveis**
- Portanto, sem recompilar o código-fonte, os mesmos bytecodes podem executar em qualquer plataforma contendo uma JVM

Fase 3: Carregando um Programa em Memória

- A JVM é invocada pelo comando `java`

Invocando a JVM para um bytecode compilado `welcome.class`
via linha de comando

```
java welcome
```

- A JVM armazena o programa na memória para executá-lo
(**carregamento**)

Fase 3: Carregando um Programa em Memória

- O **carregador de classe** da JVM pega os arquivos `.class` que contêm os bytecodes do programa e transfere-os para a memória primária
- O carregador de classe também carrega qualquer arquivo `.class` que seja chamado dentro do arquivo `.class` disparado na JVM
- Os arquivos `.class` podem ser carregados a partir de um disco em seu sistema ou por meio de uma rede

Fase 4: Verificação do Bytecode

- Enquanto as classes são carregadas, o **verificador de bytecode** examina seus bytecodes
- A verificação visa assegurar que eles são válidos e não violam restrições de segurança do Java
- Java certifica-se que programas não danifiquem o sistema (ex: vírus) além de verificar a corretude das classes

Fase 5: Execução

- A JVM interpreta e executa os bytecodes, realizando assim as ações especificadas pelo programa
- Durante a interpretação, a JVM procura pelos **hot spots**, que são partes dos bytecodes que executam com frequência
- Para os hot spots, um **compilador Just-In-Time (JIT)** (ou **compilador Java HotSpot**) traduz os bytecodes para linguagem de máquina
- Quando a JVM encontra novamente essas partes compiladas, o código de linguagem de máquina mais rápido é executado

Fase 5: Execução

- Portanto, os programas Java na realidade passam por **duas fases de compilação**:
 - 1 O código-fonte é traduzido em *bytecodes* (para a portabilidade entre JVMs em diferentes plataformas de computador)
 - 2 Durante a execução, os *bytecodes* são traduzidos em linguagem de máquina para o computador em que o programa é executado

Código Fonte vs *Bytecodes*

Código-fonte (.java)

```
public class Onibus {  
    String nome;  
    String linha;  
}  
  
class Teste {  
    public static void main(String args) {  
        Onibus o = new Onibus();  
        o.linha = "Jabaquara-Liberdade";  
    }  
}
```



Bytecodes (.class)

Compiled from "Onibus.java"
class Teste {

```
    public static void main(java.lang.String);  
    Code:  
    0: new           #2  // class Onibus  
    3: dup  
    4: invokespecial #3  Onibus."<init>":()V  
    7: astore_1  
    8: aload_1  
    9: ldc           #4  // String Jabaquara...  
   11: putfield      #5  
       // Field Onibus.linha:Ljava/lang/String;  
   14: return  
}
```

- Bytecodes são comandos que a JVM entende
- Cada comando possui exatamente **1 byte**

Material Complementar

- **Página:** Programação Orientada a Objetos: por que aprender isso?

[http://www.dicasdeprogramacao.com.br/
programacao-orientada-objetos-por-que-aprender-isso/](http://www.dicasdeprogramacao.com.br/programacao-orientada-objetos-por-que-aprender-isso/)

- **Vídeo:** Curso de Java #01 - História do Java - Gustavo Guanabara

[https:
//youtu.be/sTX0UEplF54?list=PLHz_AreHm4dkI2ZdjTwZA4mPMxWTfNSpR](https://youtu.be/sTX0UEplF54?list=PLHz_AreHm4dkI2ZdjTwZA4mPMxWTfNSpR)

- **Vídeo:** Curso de Java #02 - Como Funciona o Java - Gustavo Guanabara

https://www.youtube.com/watch?v=v_ZCtgwbS3o

Material Complementar

- List of programming languages by type

https://en.wikipedia.org/wiki/List_of_programming_languages_by_type

- Vídeo:** Curso de Java #03 - Instalando o JDK no seu Computador - Gustavo Guanabara

https://www.youtube.com/watch?v=yWU5bm_pZzY&list=PLHz_AreHm4dkI2ZdjTwZA4mPMxWTfNSpR&index=5

- Os 4 pilares da Programação Orientada a Objetos

<https://www.devmedia.com.br/>

[os-4-pilares-da-programacao-orientada-a-objetos/9264](https://www.devmedia.com.br/os-4-pilares-da-programacao-orientada-a-objetos/9264)

Material Complementar

- PATH and CLASSPATH

`https:`

`//docs.oracle.com/javase/tutorial/essential/environment/paths.html`

- Paradigma de programação

`https://pt.wikipedia.org/wiki/Paradigma_de_programa%C3%A7%C3%A3o`

Imagem do Dia [1]

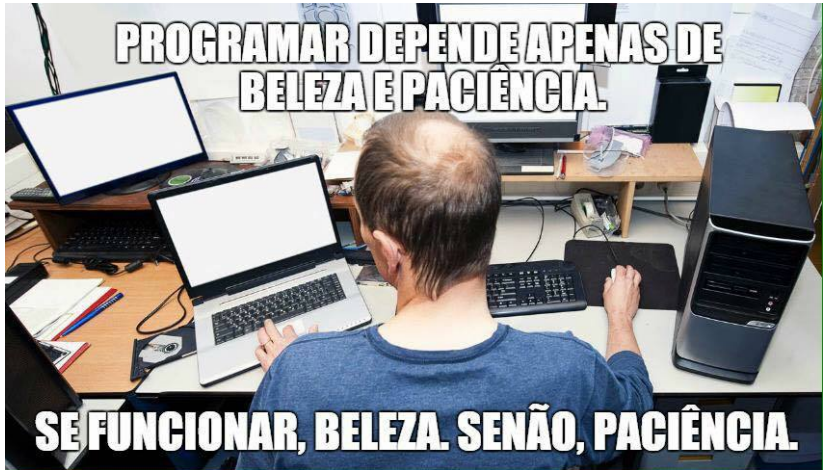


Imagem do Dia [2]



Universiotariano
@universiotarito



"Qual foi a maior loucura que você já fez
na faculdade?"

a matrícula

Programação Orientada a Objetos

<http://lives.ufms.br/moodle/>

Rafael Geraldeli Rossi
rafael.g.rossi@ufms.br

Slides baseados em [Deitel and Deitel, 2010]

Referências Bibliográficas I



Canaltech (2014).

Java lidera ranking das linguagens de programação mais utilizadas no mundo.



Deitel, P. and Deitel, H. (2010).

Java: How to Program.

How to program series. Pearson Prentice Hall, 8th edition.