



Análise Orientada a objetos

Aula 1

Prof. Me. Juliana Costa-Silva



Na aula de hoje...

1. Apresentação
2. Linguagens de programação
3. Atividade I
4. História da linguagem Java
5. Java
6. Máquina Virtual Java
7. Plataforma



Sobre a Disciplina

Programação I

Atenção!

- Quarta-feira (19 as 20:35 hs e 20:50 as 22:00 hs) - 4 aulas;
- 80 hs semestre;
- Frequência mínima 75% (Máximo de 20 faltas);



Disciplina

Objetivos da Disciplina:

1. Histórico e cenário atual da POO;
2. Programação estruturada e POO;
3. Paradigma de programação orientada a objetos;
4. Classes, Objetos;
5. Polimorfismo;
6. Sobrecarga de Métodos;
7. Herança;
8. Encapsulamento;
9. Orientação a Objetos com Java;



Como serei avaliado?

As avaliações serão feitas da seguinte forma:

- Avaliação teórica + Avaliação prática;

Prova prática: Através de um problema proposto, será possível avaliar se o aluno desenvolveu as habilidades trabalhadas na disciplina;



Dúvidas?





Compilação vs. Interpretação

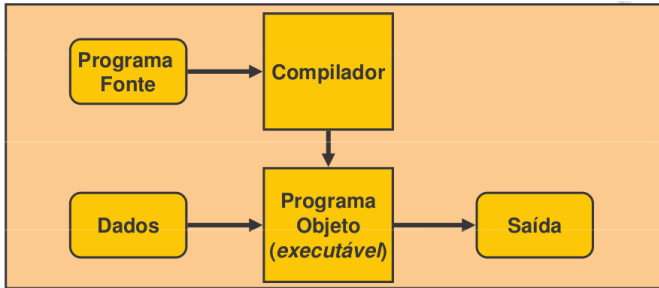
Existem duas formas básicas de implementação de uma linguagem de programação:

- Compilação;
- Interpretação;



Compilação

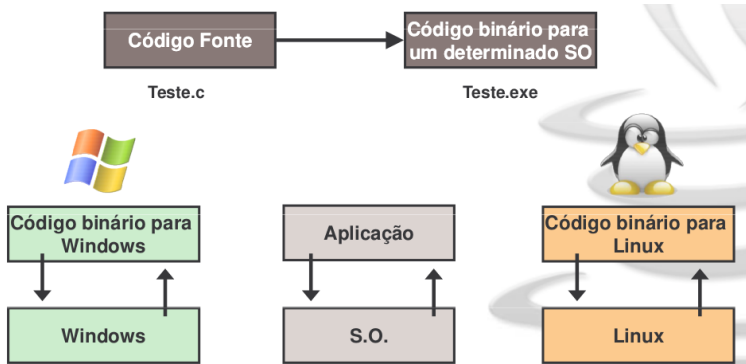
- Traduz programas escritos na linguagem de programação para linguagem objeto da máquina alvo (executável).





Compilação

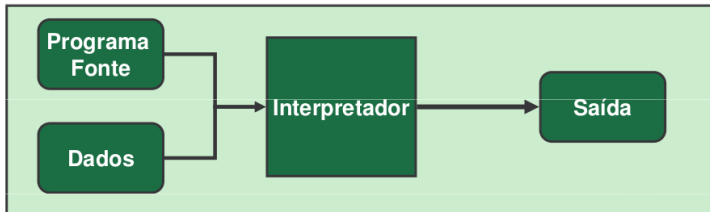
- O código fonte é compilado para uma plataforma e sistema operacional específico.





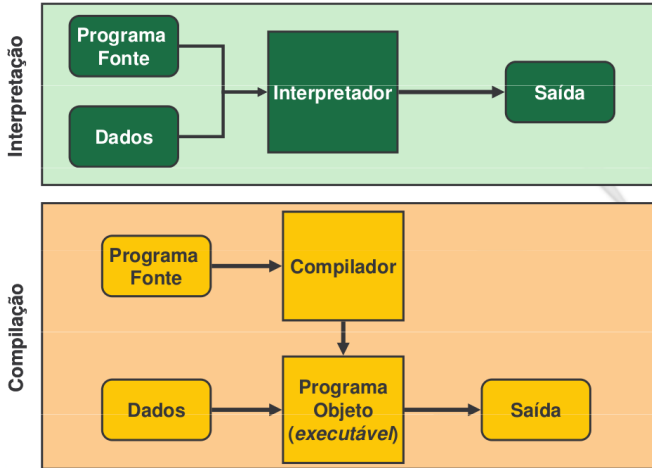
Interpretação

- Examina o programa fonte, e executa cada instrução;
- A Interpretação é feita instrução a instrução;





Compilação vs Interpretação





Malhação cerebral

1. Escreva com suas palavras o processo de compilação e interpretação.
2. Escreva quais as vantagens e desvantagens de cada método.



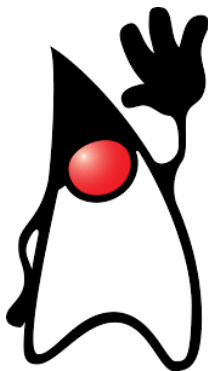
Java, como começou...

- No ano de 1991 a Sun financiou um projeto de pesquisa interno de codinome Green, chefiado por James Gosling e que criou uma linguagem baseada em C e C++ chamada Oak (carvalho) para dispositivos eletrônicos inteligentes.
- Como o projeto não alcançou os resultados de mercado esperados, e em 1993 a World Wide Web explodiu em popularidade nos EUA, a Sun resolveu adaptar o projeto para ser usado no desenvolvimento de conteúdo dinâmico para Web.



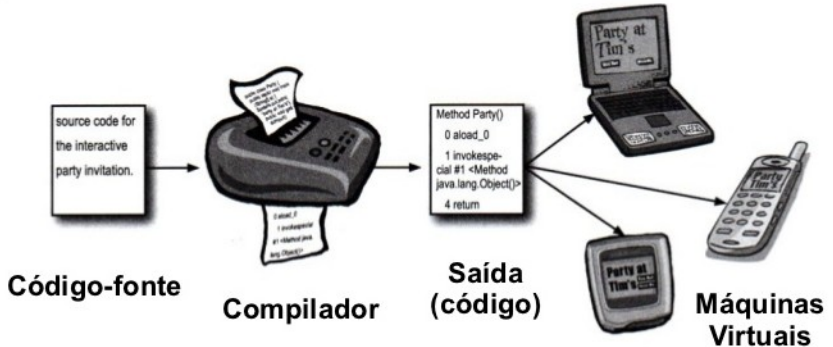
Java, como começou...

- Como já existia uma linguagem chamada Oak o nome foi trocado por Java (em homenagem a um tipo de café importado que a equipe tomava numa cafeteria perto da empresa);
- Em 23 de maio de 1995 a Sun apresentou oficialmente a linguagem Java ao mercado, na conferência SunWorld. E o seu mascote duke.



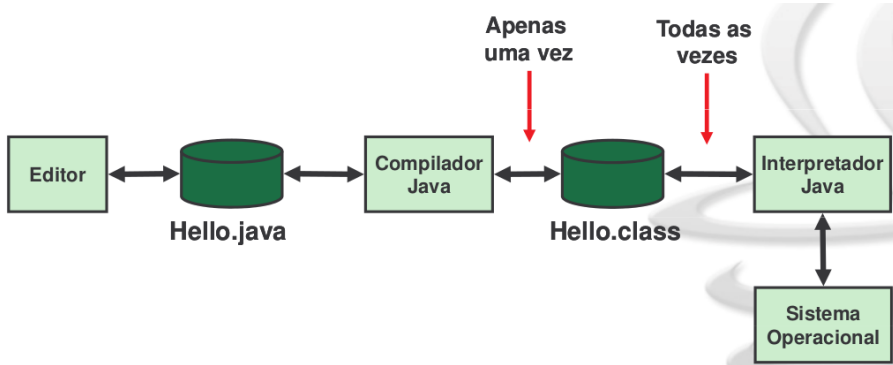


Como java funciona?





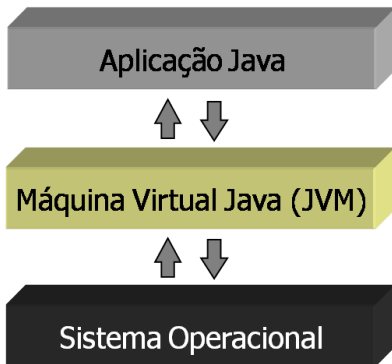
Fases de um programa Java





Máquina Virtual

A arquitetura da Linguagem Java se baseia na existência de uma máquina virtual (JVM) que deixa transparente para a aplicação a existência do Sistema Operacional e do Hardware.

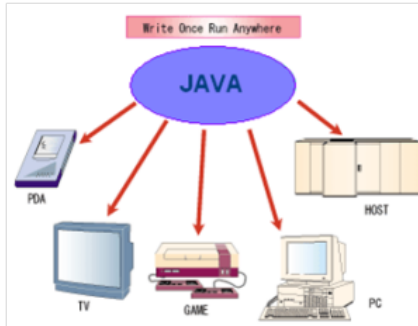




Máquina Virtual

A arquitetura da Linguagem Java se baseia na existência de uma máquina virtual (JVM) que deixa transparente para a aplicação a existência do Sistema Operacional e do Hardware.

Qual a vantagem disso??





Máquina Virtual

“Máquina Virtual” que roda as aplicações Java:

- Funciona como um interpretador;
- Checagem de erros em tempo de execução;
- Garbage Collector (GC) - Coletor de lixo;
- Torna a linguagem independente de plataformas (Windows, Linux, MacOS, ...);
- “Write once. Run anywhere”.



Tecnologia Java

- Um ambiente de aplicação e distribuição
 - ❑ Java Development Kit (JDK): conjunto de bibliotecas, classes e ferramentas usados por programadores no desenvolvimento de aplicações Java.
 - ❑ Java Runtime Environment (JRE): conjunto de bibliotecas, classes e interpretadores para execução de aplicações Java pelos usuários.
 - ❑ Documentation (docs): documentação em formato HTML das classes usadas no desenvolvimento de aplicações que vem integradas no JDK.



Tecnologia Java

- **Java Standard Edition (JSE):** versão destinada ao desenvolvimento de aplicações para desktop e estações de trabalho. Possui as APIs (Bibliotecas) essenciais a qualquer aplicação Java.
- **Java Enterprise Edition (JEE):** versão destinada ao desenvolvimento de aplicações de grande porte para rodarem em servidores de aplicações. Visa escalabilidade, robustez e segurança.
- **Java Micro Edition (JME):** versão destinada ao desenvolvimento de aplicações para dispositivos limitados em memória e processamento como: celulares, PDA e set-top boxes.
- **Java Card:** versão destinada ao desenvolvimento de aplicações para dispositivos MUITO limitados em memória e processamento, como os cartões inteligentes (Smart Cards).



Arquivos Java

- **.java** : arquivo fonte de programas criado em formato de texto puro. Pode ser manipulado em qualquer editor de texto.
- **.class** : arquivo compilado java. No formato binário (bytecodes) que pode ser executada em qualquer JVM compatível.
- **.jar** : (Java Archive) agrupamento de classes java que fazem parte de um mesmo pacote ou projeto. Podem estar compactadas (Padrão Zip).
- **.jad** : (Java Archive Descriptor). Arquivo texto que descreve o conteúdo de um arquivo .jar



Malhação Cerebral

1. Descreva com detalhes as fases do desenvolvimento de um programa em JAVA.
2. Escreva sobre a máquina virtual.



O que cada linha de código faz?

```
01  int tamanho = 27;
02  String nome = "Rex";
03  Cao meuCao = new Cao(nome, tamanho);
04  x = tamanho - 5;
05  if (x < 15) meuCao.latir(8);
06
07  while (x > 3) {
08      meuCao.brincar();
09  }
10
11  int[] numList = {2,4,6,8};
12  System.out.print("Olá");
13  System.out.print("Cão: " + nome );
14  String num = "8";
15  int z = Integer.parseInt(num);
16
17  try{
18      readTheFile("meuArquivo.txt");
19  }
20  catch (FileNotFoundException ex) {
21      System.out.print("Arquivo não encontrado");
22  }
```




O que cada linha de código faz?

01	Declara uma variável do tipo inteiro chamado ' <i>tamanho</i> ' e lhe atribui o valor 27
02	Declara uma variável de string de caracteres chamada ' <i>nome</i> ' e lhe atribui o valor " <i>Rex</i> "
03	Declara a nova variável de tipo <i>Cao</i> chamada ' <i>meuCao</i> ' e cria o novo objeto <i>Cao</i> usando ' <i>nome</i> ' e ' <i>tamanho</i> '
04	Subtrai 5 de 27 (valor de ' <i>tamanho</i> ') e atribui o valor a uma variável chamada ' <i>x</i> '
05	Se x (valor = 22) for menor do que 15, informa ao objeto <i>Cao</i> chamado ' <i>meuCao</i> ' para latir 8 vezes
06	
07	Mantém o loop até x ser maior que 3...
08	Pede ao objeto <i>Cao</i> chamado ' <i>meuCao</i> ' para brincar (independente do que isso signifique para um cão)
09	Final do loop – tudo que estiver entre { } será feito no loop
10	
11	Declara uma lista de variável do tipo inteiro 'numList' e insere 2,4,6,8
12	Exibe "Olá"... na linha de comando
13	Exibe "Cão : Rex" na linha de comando
14	Declara a variável de string de caracteres 'num' e lhe atribui o valor "8"
15	Converte a string de caracteres "8" no valor numérico 8
16	
17	Tenta fazer algo... Pode ser que o que estamos tentando não funcione
18	Lê um arquivo de texto chamado "meuArquivo.txt" (ou pelo menos tenta ler o arquivo...)
19	Fim das 'tentativas'
20	Aqui é onde você saberá se o que tentou não funcionou...
21	Se o que tentou não deu certo, exibiremos "Arquivo não encontrado" na linha de comando
22	Tudo que se encontra entre { } é o que deve ser feito se a 'tentativa' não funcionar



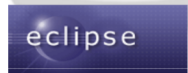
O programa JAVA

- Em Java tudo fica em uma classe;
- Geralmente cada classe fica em um arquivo (padrão de desenvolvimento);
- O arquivo deve ter o mesmo nome da classe com a extensão .java;
- Ao executar um programa, você estará executando uma classe;
- Java é *sensitive case*: **Java** != java



Definições

- Console
 - Onde se executa os comandos do SO.
Exemplo: Prompt de comandos (Windows)
- Editor de texto
 - **Exemplo:** Notepad (Windows)
- Ambiente de Desenvolvimento Integrado (IDE):
É um aplicativo que provê: um editor de códigos, um compilador e/ou interpretador, depurador.
 - **Exemplo:** NetBeans, Eclipse





Primeiro programa em Java

1. Iniciar o editor de texto;
2. Escrever o código utilizando o editor;
3. Salvar o código com o nome `cod/Ola.java`
4. Abrir o console
5. Entrar na pasta onde o arquivo de código fonte (.java) esta salvo
6. Compilar o código fonte: `javac Ola.java`
7. Executar o programa: `java Ola`



Primeiro programa Java

[linrange=1-5]./codigo/Ola.java

- `cd C:< caminho – pasta >`
- `javac Ola.java`
- `java Ola`



O método main()

- O método main() é onde a execução inicia
- Executar um programa, é o mesmo que dizer a Máquina Virtual (JVM) para carregar a classe e executar o método main()
- O método main() sempre é chamado de forma automática
- Nem toda classe exige um método main()
- Não importa quantas classes seu programa possui
- Mas deverá possuir apenas um método main() - desse modo o programa torna-se “executável”



O método main()

Exemplos:

- O parâmetro do método main é um vetor formado por textos passados via linha de comando
- O parâmetro pode ser declarado de formas diferentes, seu **nome não importa**, porém, obrigatoriamente deve ser um array de String



O método main()

[linrange=1-5]./codigo/Main.java

- Qualquer distorção neste cabeçalho, a JVM não identificará o método e ocorrerá um erro durante a execução da classe.



Saída de dados

- System.out...
- O método **print** imprime um conjunto de dados e posiciona o cursor no final deles
- O método **println** imprime um conjunto de dados e posiciona o cursor na próxima linha
- **Ambos são métodos sobrecarregados**, ou seja, existem várias versões destes métodos (funcionam para diferentes tipos e quantidades de parâmetros)



Saída padrão (print e println)

```
1 public class ExemploPrint{
2     public static void main(String[] args) {
3         String texto = "Disciplina: Programacao I";
4         int ano = 2018;
5         System.out.print(texto);           //Imprime texto sem pular linha
6         System.out.print(" - Unicesumar "); //Imprime texto sem pular linha
7         System.out.print(ano);             //Imprime um valor inteiro
8         System.out.println();              //Pula uma linha
9         System.out.println("Prof. Juliana");//Imprime o texto e pula linha
10        System.out.print("=====");
11    }
12 }
```

Saída

```
Disciplina: Programacao I - Unicesumar 2018
Prof. Juliana
=====
```



Exemplo 2 (print e println)

[linrange=1-7]cod/ExemploPrint2.java

Saída

```
Disciplina: Programacao I - Unicesumar 2018  
Prof. Juliana
```

```
=====
```



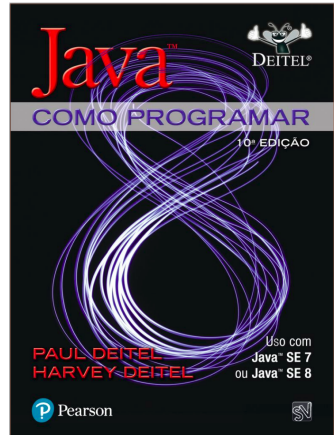
print e println

- Conseguimos a mesma saída;
- Com uma menor quantidade de comandos **print** e **println**;
- Isso foi possível por causa do uso de dois operadores;
- Operador de concatenação “+”;
- Operador de escape “*n*”.



Leitura complementar

Para mais informações sobre JAVA, leia:



Capítulo 1: [Deitel, 2016]

DISPONÍVEL NA BIBLIOTECA VIRTUAL



Referências



Deitel, Paul J.; Deitel, H. M. (2016).

Java 8: Como programar. 10ª Edição.

Pearson.