



UNAMA

50
anos



Programação Orientada a Objetos e Estrutura de Dados

Apresentação do professor:

Nome: Filipe C. Fernandes

E-mail: 040602182@prof.unama.br

Lattes: <https://lattes.cnpq.br/2458124137052507>

- ☐ Graduado em Engenharia Elétrica – UFPA
- ☐ Mestrado em Eng. Elétrica - UFPA
- ☐ Doutorando na área de Telecomunicações - UFPA

Informações gerais:

Unidade: UNAMA ALCINDO CACELA

Nome da Disciplina: POO e ED

Código da Disciplina: GSER2057665

Curso: ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Carga horária: 80h

Turno: Matutino

Semestre: 2024.2

Dia e horário

Sexta – 8h às 11h30

Professor: Filipe Cavalcanti Fernandes

E-mail: 040602182@prof.unama.br

Data de início: 09/08/2024

Data de término: 18/12/2024

Ementa:

- Conceitos básicos e terminologias de Programação orientada a objetos.
- Classe, Objetos, Atributos, Métodos, Construtores e sobrecarga.
- Instanciação e Referência de objetos, abstração ,encapsulamento e herança.
- Classes abstratas e Interfaces.
- Relacionamento entre classes e polimorfismo.
- Tratamento de exceções.
- Representação e manipulação de estruturas de dados estáticas e dinâmicas, tais como: matrizes, listas encadeadas, pilhas, filas e árvores.
- Modelagem física e meios de armazenamento.
- Organização primária de arquivos: sequencial, sequencial indexado, indexado, direto, direto calculado (hash). Ordenação de arquivos.

Objetivos de aprendizagem:

- ☐ Entender a programação orientada a objetos;
- ☐ Realizar a implementação de classes, a criação de objetos e o envio de mensagens;
- ☐ Saber utilizar a hierarquia de classes e polimorfismo em Java;
- ☐ Entender o mecanismo de tratamento de exceções em Java;
- ☐ Estruturar dados dentro de uma aplicação.

Conteúdos de aprendizagem:

Unidade I

- 1.1 Introdução à Orientação a Objetos
- 1.2 Linguagens típicas Orientadas a Objetos
- 1.3 Programação Orientada a Objeto
- 1.4 Conceitos básicos e Terminologias
- 1.5 Classes, Objetos, Atributos, Métodos, Construtores e Sobrecarga
- 1.6 Instanciação e referência de Objetos
- 1.7 Envio de Mensagens
- 1.8 Ciclo de vida de um Objeto
- 1.9 Abstração e encapsulamento

Conteúdos de aprendizagem:

Unidade II

2.1 Herança

2.2 Criação e Uso de Hierarquia de Classes

2.3 Relacionamento entre Classes

2.4 Polimorfismo

2.5 Ligação Dinâmica (Dynamic Binding)

2.6 Tratamento de Exceções

Conteúdos de aprendizagem:

Unidade III

- 3.1 Listas: Estática Sequencial / Encadeada
- 3.2 Listas: Dinâmica Simples / Encadeada
- 3.3 Listas: Aplicações

Unidade IV

- 4.1 Filas
- 4.2 Pilhas
- 4.3 Árvores
- 4.4 Técnicas de ordenação
- 4.5 Técnicas de pesquisa

Metodologia de ensino:

- Aulas presenciais expositivas-dialogadas.
- Leitura e debate de textos teóricos.
- Seminários.
- Dinâmica de grupos.
- Aulas práticas.

Avaliação:

AV1 – 04/10

- Prova: 5,0
- Apresentação de Trabalho: 5,0

AV2 – 29/11

- Prova: 5,0
- Apresentação de Trabalho : 5,0

2ª chamada 06/12

Prova Final 13/12

Avaliação:

Os critérios de avaliação seguirão o padrão determinado pelo Regimento Geral da UNAMA - Capítulo IV e Regulamento MIDDI - Subseção 6.9.

A média de aprovação discente é de 7,0 (sete) pontos;

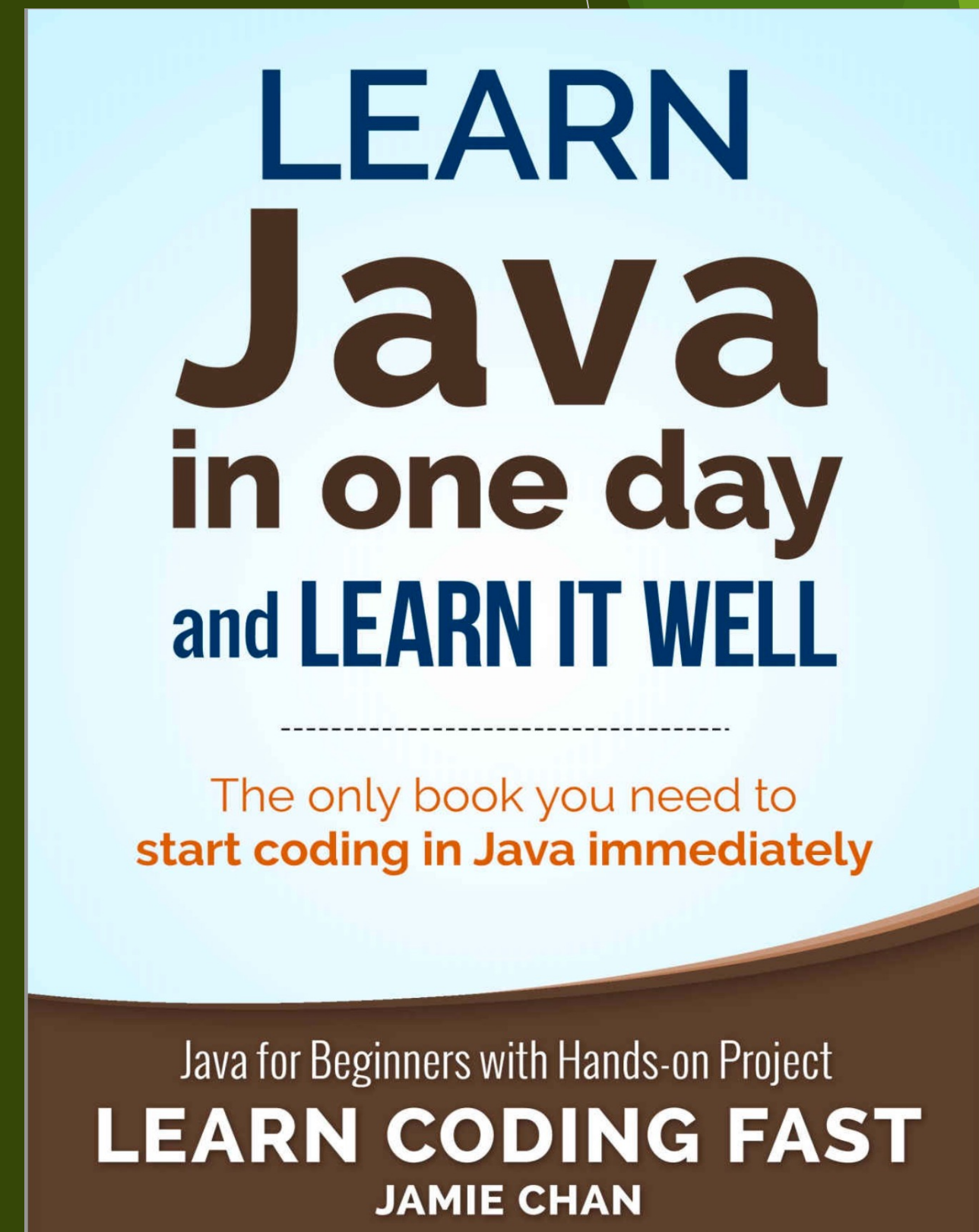
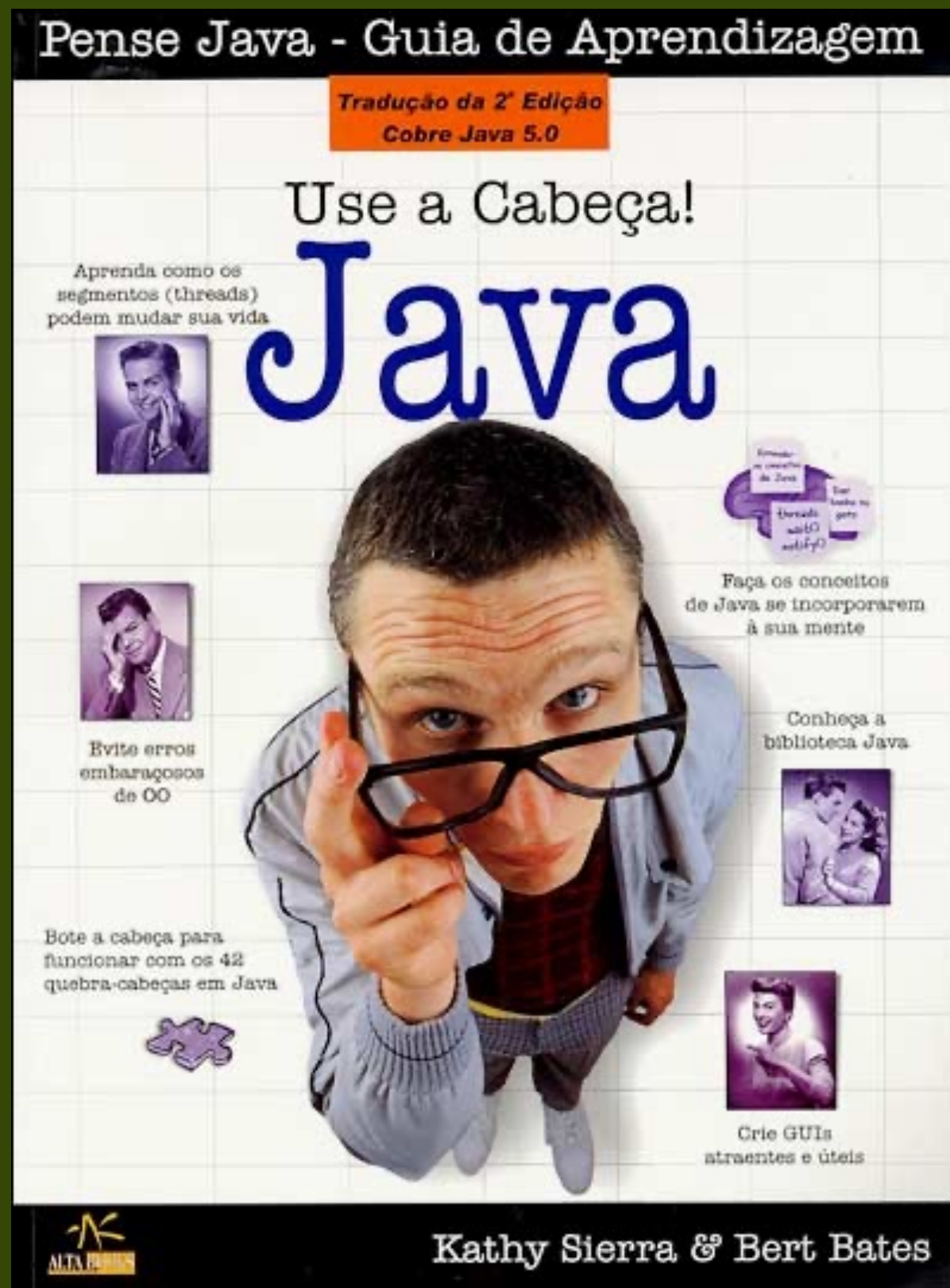
Só poderão realizar a prova final os discentes com média a partir de 4,0 (quatro) pontos e menor que 7,0 (sete) pontos;

A frequência mínima de 75% (setenta e cinco por cento) da disciplina;

Média inferior a 4,0 (quatro) é considerado como reprovação.

Cálculo de média: $AV1 + AV2 / 2$

Bibliografia da disciplina:





Introdução a Programação Orientada a Objetos

Nome: Filipe C. Fernandes

E-mail: 040602182@prof.unama.br

Introdução

□ Pré-requisitos

□ Programação |||||

□ Esforço Próprio |||||

□ Paciência ||||

□ Teoria |||

□ Para que serve esta disciplina?

□ Programação de sistemas em geral:

□ Desktop

□ Web

□ Games ...

Introdução

❑ Javadoc

- ❑ A documentação oficial da linguagem sempre é rica e completa

<https://docs.oracle.com/javase/8/docs/technotes/tools/windows/javadoc.html>



UNAMA

50
ANOS

Introdução

- ❑ Antes de começar, observações importantes:
 - ❑ Não seja fanático(a) por uma tecnologia específica (hype);
 - ❑ Aprenda sobretudo os conceitos e a serventia das ferramentas que você pode usar na implementação dos projetos;
 - ❑ Desenvolva a sua capacidade de abstração de problemas;
 - ❑ Tenha disciplina;
 - ❑ **Divirta-se programando!**



UNAMA

50
ANOS

Primeiro Programa

- ❑ Antes de começar, observações importantes:
 - ❑ Não seja fanático(a) por uma tecnologia específica (hype);
 - ❑ Aprenda sobretudo os conceitos e a serventia das ferramentas que você pode usar na implementação dos projetos;
 - ❑ Desenvolva a sua capacidade de abstração de problemas;
 - ❑ Tenha disciplina;
 - ❑ **Divirta-se programando!**



UNAMA

50
ANOS

Problemas da década de 1990...

- ▶ Gerenciamento de memória;
- ▶ Organização;
- ▶ Falta de bibliotecas;
- ▶ Reescrever parte do código ao mudar de sistema operacional;
- ▶ Custo financeiro de usar determinadas tecnologias.



História

- ▶ Ser usada em pequenos dispositivos como TVs, vídeo-cassete, liquidificador, aspirador, etc;
- ▶ Mas o lançamento foi através de um navegador de internet (Netscape) para rodar pequenas aplicações (*applets*);
- ▶ Apesar de ter sido idealizado com um propósito e lançado com outro, o Java ganhou destaque no lado do servidor.



Java

História

- ▶ Foi criada por um grupo de engenheiros da Sun Microsystems (o “Green Team”) na década de 1990 -> inovação tecnológica;
- ▶ Na década de 2010 a empresa Oracle comprou a Sun Microsystems, fortalecendo ainda mais a sua marca.
- ▶ História da linguagem Java em:
<http://www.java.com/en/javahistory/>
- ▶ Vídeo interessante: <http://tinyurl.com/histjava> /
<https://youtu.be/WAy9mgEYb6o>



UNAMA

50
ANOS

Java

História - No Brasil e no Pará

Diversos grupos de usuários se formaram para disseminar o conhecimento e trocar dúvidas sobre Java:

- ▶ GUJ - <http://www.guj.com.br>
- ▶ SouJava - <http://soujava.org.br/>
- ▶ JavaFree (UOL) - <http://javafree.uol.com.br/>
- ▶ TáSafo (comunidade paraense) - <https://tasafo.wordpress.com/>



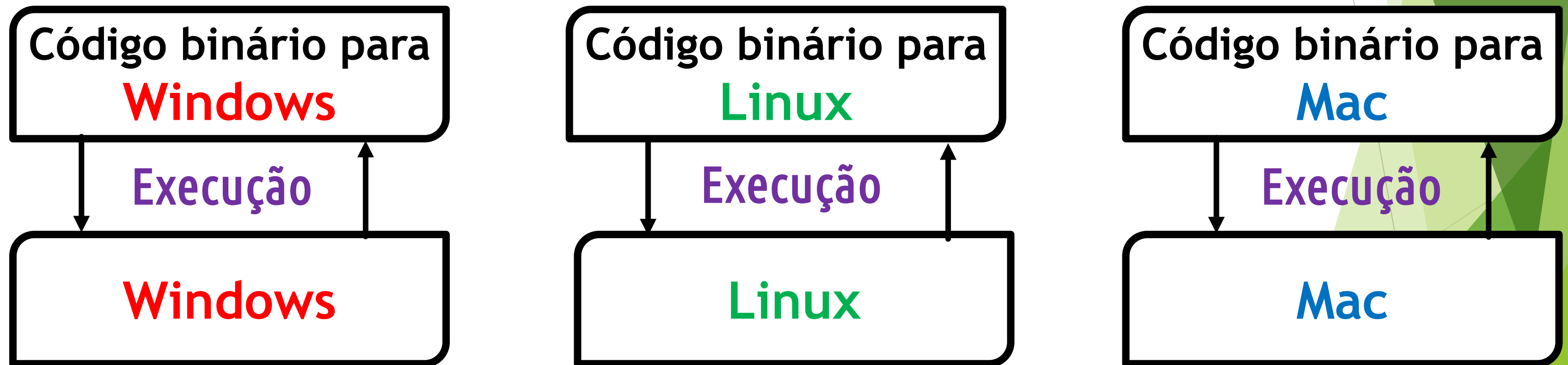
UNAMA

50
ANOS

Java

Máquina Virtual

- Linguagens de programação como C e Pascal: o código criado pelo programador é compilado para código de máquina específico de uma plataforma/sistema operacional.

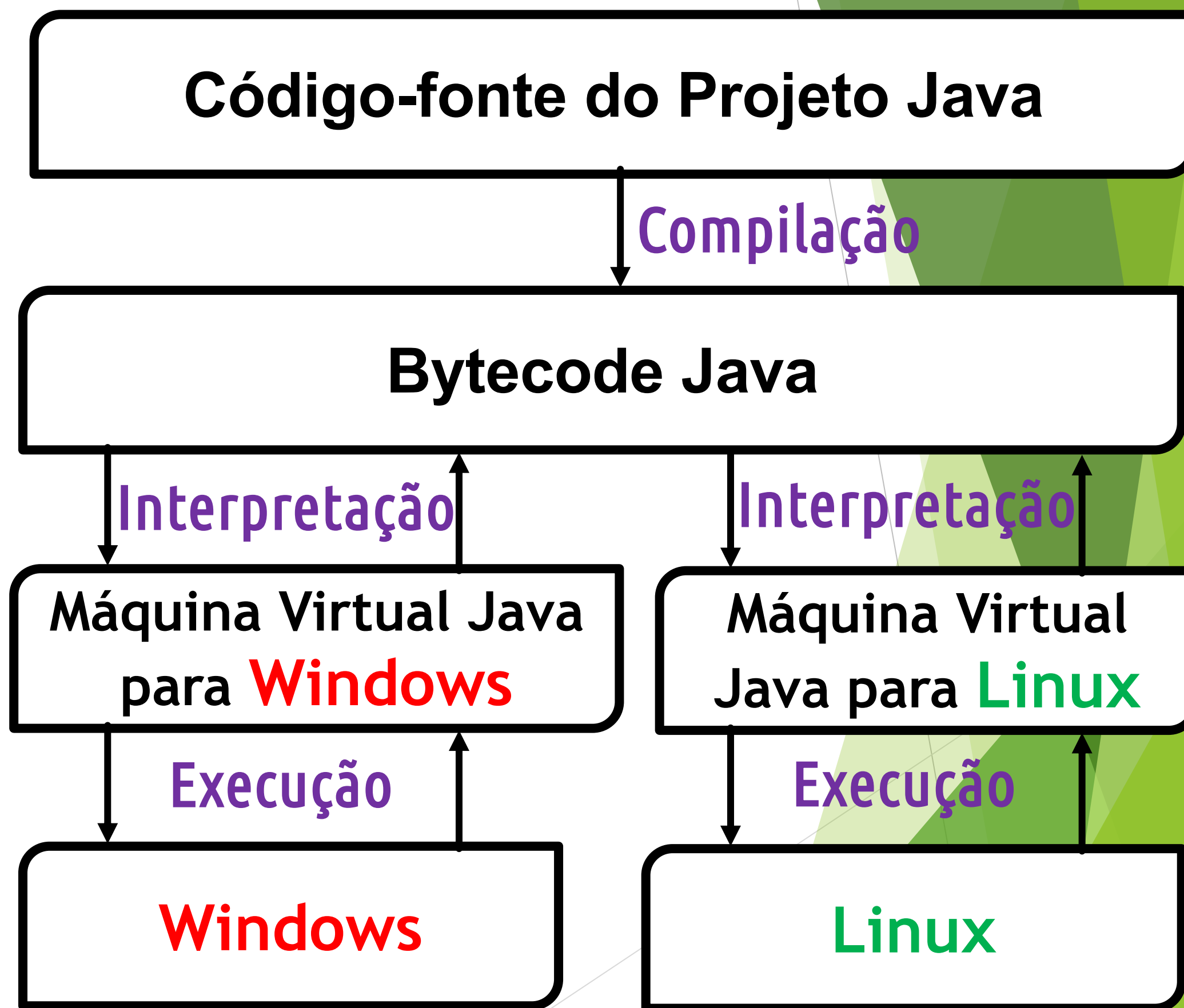


Ex. de Programas: Firefox, OpenOffice, Dropbox...

Java

Máquina Virtual

- ▶ Em Java: existe a máquina virtual que é uma camada extra entre o código binário (*Bytecode*) e o Sistema Operacional.
- ▶ Essa camada é responsável por fazer a inter-pretação do bytecode.



Exemplo de programa simples (arquivo OlaMundo.java)

Pacote Classe Principal

```
package olamundo;  
public class OlaMundo{ // Inicio da classe  
    public static void main(String args[]) { // Método principal  
  
        System.out.println("Hello Java");  
  
    } // Fim do Método principal  
} // Fim da classe
```



UNAMA

50
ANOS

Máquina Virtual Java (MVJ) – Vantagens

- ▶ Também conhecida como JVM (*Java Virtual Machine*)
- ▶ Permite a independência de Sistema Operacional (SO);
- ▶ Alocação de memória facilitada;
- ▶ Se a aplicação Java trava, não afeta o SO; mas só a máquina virtual;
- ▶ A máquina virtual não interpreta JAVA diretamente e sim o código Java compilado (o *bytecode*).



UNAMA

50
ANOS

Máquina Virtual – Desvantagens

- ▶ Performance prejudicada
- ▶ Mas isso tem mudado muito:
 - ▶ Compilador JIT (*Just inTime Compiler*): para melhorar a performance, o compilador detecta pontos do programa que estão sendo executados com bastante frequência e os compila para instruções nativas da plataforma. -> Exemplo: laços de repetição;
 - ▶ O poder de processamento dos dispositivos atuais é maior.

Máquina Virtual – Outras informações

- ▶ Existem outras especificações de JVM;
- ▶ A especificação oficial é Oracle JVM (site da Oracle);
- ▶ Exemplo de outras: JRockit da BEA, J9 da IBM;
- ▶ Caso uma empresa precise adequar o interpretador de bytecode, ela pode criar sua própria máquina virtual;
- ▶ Todas as JVM são certificadas pela Oracle -> Motivo: garantia de compatibilidade.



UNAMA

50
ANOS

JVM? JRE? JDK? O que são afinal?

- ▶ **JVM:** *Java Virtual Machine* (Máquina Virtual Java).
- ▶ **JRE:** *Java Runtime Environment* (Ambiente de Execução Java)
-> formado pela JVM e bibliotecas.
- ▶ **JDK:** *Java Development Kit* (Kit de Desenvolvimento Java)
-> formado pela JRE e ferramentas (compilador, documentação).

*O Netbeans e o Eclipse geralmente já vem junto com o JDK.



UNAMA

50
ANOS

Onde usar e os objetivos da linguagem

- ▶ Projetos de médio a grande porte, com várias pessoas trabalhando ao mesmo tempo;
- ▶ Projetos que mudam e crescem com o passar do tempo. Legibilidade garantida no futuro;
- ▶ Manutenções no futuro serão mais fáceis e rápidas (desde que boas práticas sejam seguidas);
- ▶ Quantidade enorme de bibliotecas GRATUITAS facilitam a programação e aumentam a produtividade.
 - ▶ relatórios, gráficos, sistemas de busca, geração de código de barra, manipulação de XML, tocadores de vídeo, manipuladores de texto, banco de dados, impressão, etc.

Onde usar e os objetivos da linguagem

- ▶ Projeto que deve funcionar em qualquer sistema operacional (interoperabilidade)
 - > *"Write once, run anywhere"* / "Escreva uma vez, rode em qualquer lugar".
- ▶ Há grande procura por profissionais de TI que dominam Java;
- ▶ Certificações em Java + inglês (TOEFL) + experiência com a linguagem = grandes chances de trabalhar em projetos importantes e receber um salário alto;
- ▶ Entre 3k e 4k (inicial) + benefícios. Depende muito da empresa.



Revisão de programação, só que com Java

Nome: Filipe C. Fernandes

E-mail: 040602182@prof.unama.br

Exemplo de programa simples (arquivo OlaMundo.java)

Pacote Classe Principal

```
package olamundo;  
public class OlaMundo{ // Inicio da classe  
    public static void main(String args[]) { // Método principal  
  
        System.out.println("Hello Java");  
  
    } // Fim do Método principal  
} // Fim da classe
```



UNAMA

50
ANOS

Variáveis e operadores

- ▶ O que são variáveis?
- ▶ Tipos primitivos em Java
 - ▶ `byte` inteiro de -128 a 127. Usa 1 byte
 - ▶ `short` Usa 2 bytes. Vai de -32768 a 32767
 - ▶ `int` Usa 4 bytes. Vai de -2^{31} a $2^{31}-1$
 - ▶ `long` Usa 8 bytes
 - ▶ `float` Usa 4 bytes e serve para ponto flutuante
 - ▶ `double` Usa 8 bytes. Ponto flutuante
 - ▶ `char` Armazena um único caractere Unicode
 - ▶ `boolean` Tipo especial de dado, 0 ou 1

Variáveis e operadores

Nomes de variáveis

- ▶ Podem conter: Letras, números _ ou \$
- ▶ Não pode começar com número
- ▶ Camel Case



UNAMA

50
ANOS

Variáveis e operadores

Operadores básicos

Suppose $x = 7$, $y = 2$

Addition: $x + y \Rightarrow 9$

Subtraction: $x - y \Rightarrow 5$

Multiplication: $x * y \Rightarrow 14$

Division: $x / y = 3$ (arredonda para o inteiro mais próximo)

Modulus: $x \% y = 1$ (retorna o resto da divisão)



UNAMA

50
ANOS

Variáveis e operadores

Type Casting em Java

- ▶ Linguagem fortemente tipada.
 - ▶ As variáveis não aceitam valores diferentes dos previamente definidos

```
short age = 10; // 2 Bytes
```

```
double myDouble = age; // 8 Bytes
```

```
int x = (int) 20.9;
```



UNAMA

50
ANOS

Variáveis e operadores

Arrays

- ▶ É um tipo de variável utilizada para armazenar um conjunto de valores

```
int[] userAge;
```

```
userAge = new int[] {21, 22, 23, 24, 25}
```



UNAMA

50
ANOS

Variáveis e operadores

Arrays

- Para termos acesso aos métodos de manipulação de arrays precisamos importar o

```
import java.util.Arrays;
```

```
int[] userAge;
```

```
userAge = new int[] {21, 22, 23, 24, 25}
```



UNAMA

50
ANOS



Desafios de Hoje

Nome: Filipe C. Fernandes

E-mail: 040602182@prof.unama.br

Exercícios

Desafio 1:

- Desenvolva um programa que leia um inteiro n e imprima n linhas na tela com o seguinte formato (exemplo se $n = 6$):

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
```



UNAMA

50
ANOS

Exercícios

Desafio 2:

- ❑ Faça uma função para calcular $n!$ para qualquer $n \geq 0$.



UNAMA

50
ANOS

Exercícios

Desafio 3:

- Sabe-se que cada número da forma n^3 é igual a soma de n ímpares consecutivos. Por exemplo:

$$1^3 = 1$$

$$2^3 = 3 + 5$$

$$3^3 = 7 + 9 + 11$$

$$4^3 = 13 + 15 + 17 + 19$$

- Dado um número inteiro m , $m > 0$, faça um programa em Java que mostre os ímpares consecutivos cuja soma é igual a n^3 , para n assumindo valores de 1 a m .



UNAMA

50
ANOS

Exercícios

Desafio 4:

- ❑ Dado um array A de tamanho N . Você precisa encontrar a soma entre os elementos máximo e mínimo desse array.

Observação: Fazer o mínimo de comparações.

Fonte: interviewbit.com

Exercícios

Desafio 4:

- Desenvolva um programa que pede para o usuário as notas de cinco alunos e que calcule a média das notas. Depois, mostre a média calculada, quais notas foram menores que a média e quais foram maiores ou iguais a média.

Exemplo:

Notas = [7.0, 8.0, 4.5, 9.0, 10.0]

Média: 7,7

Notas menores que a média: 7.0, 4.5

Notas maiores que a média: 8.0, 9.0, 10.0