

# Orientação a Objetos com Java

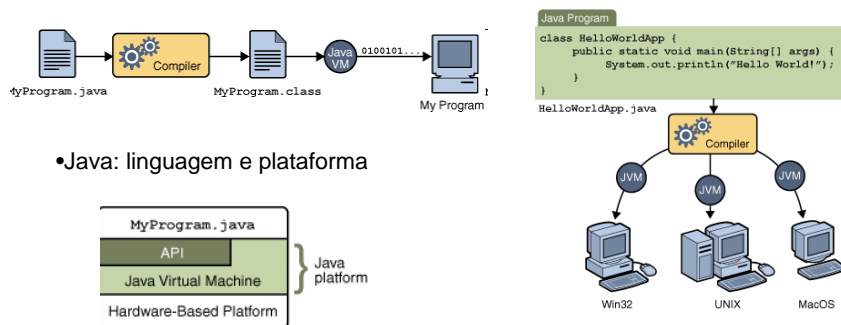
prof. Angelo C. Loula

# Tecnologia Java

- Java
  - uma linguagem orientada a objetos
  - neutra em relação a arquitetura de hardware (JVM)
  - sintaxe próxima ao C/C++
  - gerência robusta de memória (sem ponteiros, e *garbage collection* automático)
  - compilada/interpretada (bytecodes)

White Paper, The Java Language Environment  
<http://java.sun.com/docs/white/langenv/>

# Tecnologia Java



•Java: linguagem e plataforma

# Rodando o Programa em Java

- Necessário:
  - JRE: Java Runtime Environment. Necessário para **rodar** as aplicações
  - JDK: Java Development Kit. Necessário para **compilar** as aplicações. *Inclui a JRE*
  - Editor: Onde as aplicações são escritas
    - Editores de Texto: Notepad, prompt do DOS, VI, VIM...
    - Ferramentas de programação: Eclipse, Netbeans

# Primeiro Programa em Java

```
// Arquivo OlaMundo.java
```

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

- Quando o interpretador Java executa uma aplicação, ele começa chamando o método *main*
- O método *main*, então, chama todos os outros métodos necessários para executar a aplicação
- Array de Strings (String [] args): permite ao programa receber parâmetros de inicialização

slide baseado em material do prof. Kellyton Brito 5

# Exemplo de Classe

```
class Bicycle {  
    int cadence = 0;  
    int speed = 0;  
    int gear = 1;  
  
    void changeCadence(int newValue) {  
        cadence = newValue;  
    }  
    void changeGear(int newValue) {  
        gear = newValue;  
    }  
    void speedUp(int increment) {  
        speed = speed + increment;  
    }  
    void applyBrakes(int decrement) {  
        speed = speed - decrement;  
    }  
    void printStates() {  
        System.out.println("cadence:"+cadence+"  
        speed:"+speed+" gear:"+gear);  
    }  
}
```

Definição da classe

Atributos

Métodos

6

# Básico do Java

- Atributos de Objetos (Instâncias de Classes)

```
class Bicycle {  
    int cadence = 0;  
    int speed = 0;  
    int gear = 1;  
}
```

- Atributos de Classe

```
class Bicycle {  
    static int numGears = 6;  
}
```

7

# Básico do Java

- Variáveis Locais

```
void changeGear() {  
    int currentGear = gear;  
    System.out.println("Gear  
    now:"+currentGear);  
}
```

- Parâmetros

```
void speedUp(int increment) {  
    speed = speed + increment;  
}
```

- Convenção de Nomes

```
NUM_WHEELS, numGears, gears
```

8

# Básico do Java

- Tipos Primitivos
  - byte: 8-bit, -128 até 127
  - short: 16-bit, -32,768 até 32,767
  - int: 32-bit, -2,147,483,648 até 2,147,483,647
  - long: 64-bit, -9,223,372,036,854,775,808 até 9,223,372,036,854,775,807
  - float: 32-bit IEEE 754 floating point (não deve ser usado para valores precisos, como dinheiro, para isso use a classe `java.math.BigDecimal`)
  - double: 64-bit IEEE 754 floating point (não deve ser usado para valores precisos)
  - boolean: `true` e `false`
  - char: 16-bit Unicode character, `'\u0000'` (ou 0) até `'\uffff'` (ou 65,535).

9

# Básico do Java

```
boolean result = true;
char capitalC = 'C';
byte b = 100;
short s = 10000;
int i = 100000;

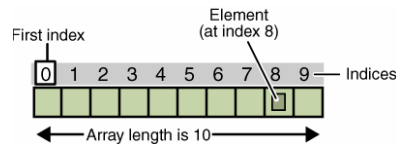
int decVal = 26; // 26 em decimal
int octVal = 032; // 26 octal
int hexVal = 0x1a; // 26 hexadecimal

double d1 = 123.4;
double d2 = 1.234e2; // notação científica
float f1 = 123.4f;
```

10

# Básico do Java

## • Arrays



```
class ArrayDemo {
    public static void main(String[] args) {
        int[] anArray; // declaração
        anArray = new int[5]; // aloca memória para 5 inteiros
        anArray[0] = 100; // inicializa primeiro elemento
        anArray[1] = 200;
        anArray[2] = 300;
        anArray[3] = 400;
        anArray[4] = 500;
        System.out.println("Element at 0: " + anArray[0]);
        System.out.println("Element at 1: " + anArray[1]);
        System.out.println("Element at 2: " + anArray[2]);
        System.out.println("Element at 3: " + anArray[3]);
        System.out.println("Element at 4: " + anArray[4]);
    }
}
```

11

# Básico do Java

## • Arrays

```
int[] anArray = {100, 200, 300, 400, 500};
```

## • Arrays de Arrays

```
int[][] matrix = {{100, 200}, {300, 400, 500}};
```

← cada linha pode ter quantidade diferente de elementos

```
System.out.println("Element at 0,1: " + matrix[0][1]);
```

```
System.out.println("Element at 1,2: " + matrix[1][2]);
```

## • Copiando Arrays

```
System.arraycopy(Object src, int srcPos,
                 Object dest, int destPos, int length);
```

12

## Básico do Java

- Operador de atribuição: =
- Operadores aritméticos: +, -, \*, /, %,
- Operadores unários: +, -, ++, --, !
- Operadores relacionais: ==, !=, >, <, >=, <=
- Operadores condicionais: &&, ||
- Comparador de tipo: instanceof
  - compara o objeto com a classe,
  - mybyke instanceof Bicycle

13

## Básico do Java

- Controle de Fluxo

```
if(condição){
    ...
} else {
    ...
}

switch(variável) {
    case valor1:
        ...
        break;
    case valor2:
        ...
        break;
    case valor3:
        ...
        break;
    default:
        ...
        break;
}
```

14

## Básico do Java

- Controle de Fluxo

```
while(condição){
    ...
}

for (init; cond; inc) {
    ...
}

do {
    ...
} while(condição);
```

15