

Aula 03

Conceitos básicos de Java

Programação III

Prof. Augusto César Oliveira

augusto.oliveira@unicap.br

Na aula passada...

- Conhecer a origem da linguagem Java;
- Identificar as principais características da linguagem Java;
- Compreender a estrutura básica de um programa Java: declaração da classe, método main e formatação do código;
- Instalar o Kit de Desenvolvimento Java (JDK) em seu sistema.

O objetivo da aula de hoje...

- Declarar variáveis corretamente;
- Identificar os diferentes tipos de dados em programação;
- Utilizar operadores de programação na resolução de problemas;
- Entender o conceito de fluxo de entrada e saída padrão em programação.

Premissas

- Parte-se do princípio que os alunos **já estão familiarizados** com conceitos como:
 1. Algoritmos;
 2. Tipos de dados, variáveis, operadores, E/S;
 3. Estruturas de decisão;
 4. Estruturas de repetição;
 5. Tipos estruturados de dados: arrays, listas.

1.

Variáveis em Java

Conceitos básicos de Java



Variáveis e identificadores

- Cada variável precisa de um **identificador único** que possa distingui-las umas das outras;
- Um identificador válido é uma **sequência de uma ou mais letras, números ou underline**;
- **Não pode haver:** espaços, símbolos ou caracteres de pontuação;
- Sempre começar com uma letra;
- Não pode ser uma **palavra reservada** da linguagem;
- Java é **case sensitive**: faz distinção entre letras maiúsculas e minúsculas;

Variáveis e tipos primitivos de dados

- Para declararmos uma variável em Java precisamos especificar o **tipo de dado** que a **variável** irá guardar.

Type	Size	Range	Default
boolean	1 bit	true or false	false
byte	8 bits	[-128, 127]	0
short	16 bits	[-32,768, 32,767]	0
char	16 bits	['\u0000', '\uffff'] or [0, 65535]	'\u0000'
int	32 bits	[-2,147,483,648 to 2,147,483,647]	0
long	64 bits	$[-2^{63}, 2^{63}-1]$	0
float	32 bits	32-bit IEEE 754 floating-point	0.0
double	64 bits	64-bit IEEE 754 floating-point	0.0

Variáveis e string

- Strings são uma **sequência de caracteres**.
- Java provê **suporte** para strings através da **classe String**.

Declaração de variáveis

- Para que uma **variável** possa ser utilizada é preciso declará-la.
- Sintaxe para **declaração** de variáveis em Java:
- Exemplos:

```
int inteiro;  
double real;  
boolean logico;  
String cadeia;
```

Atribuição de variáveis

- Quando declaramos uma variável seu valor padrão é **indeterminado**.
- Se tentarmos usá-la antes de atribuirmos um valor possivelmente teremos um erro!
- Sintaxe para **atribuição de variáveis**:

```
tipo identificador = valor;
```

```
identificador = valor;
```

```
int inteiro = 10;
```

```
int inteiro;  
inteiro = 10;
```

Exemplo de uso de variáveis | int

```
public class Variaveis {  
  
    public static void main(String[] args) {  
        int numero1 = 10;  
        int numero2 = 20;  
  
        int resultado = numero1 + numero2;  
  
        System.out.println("O resultado é: " + resultado);  
    }  
  
}
```

Exemplo de uso de variáveis | String

```
public class Variaveis {  
  
    public static void main(String[] args) {  
        String nome = "Bruno";  
        String sobrenome = "Cartaxo";  
  
        System.out.println("O seu nome é: " + nome + sobrenome);  
    }  
}
```

2.

Tipos de operadores

Conceitos básicos em Java

Operadores

- A partir do momento que possuímos **variáveis** e **valores** podemos realizar **operações** com elas.
- Principais operadores em Java:

Operadores	Grupo	Exemplos de uso
<code>+, -, *, / , %</code>	Operadores aritméticos	<code>a + b; 1 - a; 10 % 2;</code>
<code>--, ++</code>	Operadores de incremento	<code>a++; a--; ++a; --a;</code>
<code>==, !=, >, <, >=, <=</code>	Operadores relacionais	<code>1==1; a != 5; 5 > 2; b <= c;</code>
<code>!, &&, </code>	Operadores lógicos	<code>true && false; a false;</code>
<code>+</code>	Concatenação de Strings	<code>"Nome" + "Sobrenome"; "Bruno" + sobrenome</code>

Exemplo de uso de operadores

```
public class Operadores {  
    public static void main(String[] args) {  
        int a;  
        int b;  
        a = 10;  
        b = 4;  
        a = b;  
        b = 7;  
        System.out.println("a: " + a);  
        System.out.println("b: " + b);  
    }  
}
```

Exemplo de uso de operadores

```
public class Operadores{  
    public static void main(String[] args) {  
        int a = 0;  
        int b = 10;  
        int c = a + b;  
        int d = c / 5;  
        System.out.println("c: " + c);  
        System.out.println("d: " + d);  
        int e = 0;  
        e++;  
        System.out.println("e: " + e);  
        int f = 5 + e++;  
        System.out.println("f: " + f);  
    }  
}
```


3.

Fluxo de entrada e saída padrão (E/S)

Conceitos básicos de Java



Entrada e saída padrão

- Permite ao usuário **interagir com o sistema**.
- Utiliza-se o método `System.out.println()` para saída padrão em tela.
- Utiliza-se o objeto `Scanner` e seus métodos para entrada padrão do teclado:
 - `nextInt()`
 - `nextDouble()`
 - `nextBoolean()`
 - `nextLine()` ou `next()` – para string

Exemplo de entrada padrão

```
import java.util.Scanner;

public class EntradaPadrao {
    public static void main(String[] args) {
        Scanner leitor = new Scanner(System.in);
        int inteiro = leitor.nextInt();
        double real = leitor.nextDouble();
        boolean logico = leitor.nextBoolean();
        String cadeia = leitor.nextLine();
    }
}
```

Exemplo de entrada padrão

```
import java.util.Scanner;
```

Para usar o objeto Scanner é preciso importar sua classe

```
public class EntradaPadrao {
```

```
    public static void main(String[] args) {
```

```
        Scanner leitor = new Scanner(System.in);
```

Criando o objeto Scanner

```
        int inteiro = leitor.nextInt();
```

```
        double real = leitor.nextDouble();
```

```
        boolean logico = leitor.nextBoolean();
```

```
        String cadeia = leitor.nextLine();
```

```
    }
```

```
}
```

Usando o objeto Scanner para ler do teclado

Exemplo de saída padrão

```
public class SaidaPadrao{  
  
    public static void main(String[] args) {  
        System.out.println(1);  
        System.out.println(1.5);  
        System.out.println(true);  
        System.out.println("Minha cadeia de caracteres");  
    }  
  
}
```

Exemplo de saída padrão

```
public class SaidaPadrao{  
  
    public static void main(String[] args) {  
        System.out.println(1);  
        System.out.println(1.5);  
        System.out.println(true);  
        System.out.println("Minha cadeia de caracteres");  
    }  
}
```

**Escrevendo na tela com o
método System.out.println()**



4.

Considerações finais

Conceitos básicos de Java



O que aprendemos hoje?

- Declarar variáveis corretamente;
- Identificar os diferentes tipos de dados em programação;
- Utilizar operadores de programação na resolução de problemas;
- O conceito de fluxo de entrada e saída padrão em programação.

Próxima aula...

-

5.

Exercício de fixação

Google Classroom

Conceitos básicos de Java

- Link da atividade: [clique aqui](#).



ATIVIDADE

Universidade Católica de Pernambuco

Professor: Augusto César Oliveira

Disciplina: Programação III / POO

Aluno(a): _____ data: __/__/__

Aula 03 - Conceitos básicos de Java

1. Solicite ao usuário **seu nome** e imprima uma **mensagem de boas-vindas** na tela.
2. Peça ao usuário que digite **sua idade** em texto (por exemplo, "18") e converta-a em um **número inteiro**.
3. Receba um **número inteiro** do usuário e converta-o em um **número decimal** (float).
4. Peça ao usuário para digitar **dois números inteiros** e exiba a **soma** deles.

Aula 03

Conceitos básicos de Java

Programação III

Prof. Augusto César Oliveira

augusto.oliveira@unicap.br