FIAP GRADUAÇÃO

Agenda

- Tipos de dados em Java
- Input (Scanner)
- Output (System.out.println)

Tipos de dados

Java é uma linguagem fortemente tipada, logo, ao declarar uma variável precisamos informar qual o tipo desta variável.

Primitivos x Referência

Tipos de dados - primitivos

Os tipos primitivos são valores simples, não têm diversos atributos e nem possuem métodos.

Tipos de dados - primitivos

- short (Inteiros entre -32.768 e 32.767)
- int (Inteiros entre -2.147.483.648 e 2.147.483.647)
- long (Inteiros entre -9.223.372.036.854.775.808 e 9.223.372.036.854.775.807)
- float (Números com ponto flutuante)
- double (Números com ponto flutuante)
- boolean (true ou false)
- char ('A', 'B', 'C', 65... Armazenam apenas um caractere entre aspas simples. Pode-se usar inteiros representando os caracteres https://unicode-table.com/en/)
- byte (Inteiros entre -128 e 127)

Tipos de dados - referência

É uma “referência” a um objeto, que pode ter diversos atributos ou métodos. Exemplo:

Pessoa -> Possui nome, idade, sexo, fala, come e bebe

Java Naming Conventions

Vamos seguir as convenções Java para que nosso código fique padronizado com o código escrito em qualquer lugar do mundo:

- Nome de classe: padrão UpperCamelCase (Ex: PessoaJuridica).
- Atributos e métodos: padrão lowerCamelCase (Ex: dataDeNascimento, iniciarPrograma())

Importante: Não utilizar espaços, caracteres especiais ou acentuação.

Boas práticas

Muitas vezes vamos escutar o termo “boas práticas” de programação. São algumas regras que, apesar de não impedirem o resultado correto do programa, tornarão o nosso código melhor

Boa prática #1

As classes, métodos e variáveis devem ter nomes significativos:

```java
int a;

LocalDate b = LocalDate.now();
LocalDate c = LocalDate.of( year: 1986,  month: 6,  dayOfMonth: 15);

a = Period.between(c, b).getYears();

System.out.println("Idade: " + a);
```

```java
int idade;

LocalDate dataAtual = LocalDate.now();
LocalDate dataNascimento = LocalDate.of( year: 1986,  month: 6,  dayOfMonth: 15);

idade = Period.between(dataNascimento, dataAtual).getYears();

System.out.println("Idade: " + idade);
```

Boa prática #2

O código-fonte deve estar identado:

```
public
class
Pessoa
{String nome;int idade;}
```

```
public class Pessoa {

    String nome;
    int idade;

}
```

Boa prática #?

Durante o curso vamos falar de uma série de boas práticas. Mas se você se interessou e quiser conhecer mais recomendo ler sobre SOLID que são uma séria de boas práticas voltadas a programação orientada a objetos.

Erros comuns

Alguns erros comuns que devemos nos atentar quando estamos programando:

- Esquecer de fechar os delimitadores (aspas, parentes, chaves).
- Esquecer ; (ponto e vírgula) no final das instruções.
- Não lembrar a localização do projeto.
- Typo -> Erros de digitação ao chamar variáveis, classes ou programas.

Erros comuns

Alguns erros comuns que devemos nos atentar quando estamos programando:

- Esquecer de fechar os delimitadores (aspas, parentes, chaves).
- Esquecer ; (ponto e vírgula) no final das instruções.
- Não lembrar a localização do projeto.
- Typo -> Erros de digitação ao chamar variáveis, classes ou programas.

Saida de dados

Inicialmente vamos utilizar o console para exibir a saída de dados dos nossos programas:

Programa

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

    System.out.println("Olá mundo");

}
```

Console

```
/Library/Java/JavaVirtualMachines/jdk1.
Olá mundo

Process finished with exit code 0
```

Dica: No IntelliJ é só digitar “sout” e dar enter

Operadores aritiméticos

Operador	Descrição
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Módulo - Resto da divisão
++	Pré-Incremento ou Pós-Incremento
--	Pŕe-Decremento ou Pós-Decremento

Operadores aritiméticos

Operador	Descrição
+=	Atribuição aditiva
-=	Atribuição subtrativa
*=	Atribuição multiplicativa
/=	Atribuição de divisão
%=	Atribuição de módulo

Exercício

Crie um programa que executa as 4 operações matemáticas básicas:

Adição, subtração, multiplicação e divisão.

Desafio #1

Defina os operandos de forma randomica utilizando a classe Random do pacote java.util

Desafio #2

Imprima se o resultado é par ou ímpar

Entrada de dados

Para que receber as entradas de dados do usuário vamos utilizar a classe Scanner:

Programa

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

System.out.println("Qual seu nome?");

String nome = scanner.next();

System.out.println("Bem vindo "+ nome);
```

Console

```
Qual seu nome?
Fabio
Bem vindo Fabio

Process finished with exit code 0
```

Importante: É necessário importar a classe scanner declarando: `import java.util.Scanner;`

Entrada de dados

O funcionamento do Scanner é “esperar” por algum input do usuário no console toda vez que for chamado algum método next(). Podemos ter next para diferentes tipos de dados:

- scanner.nextByte(); //para ler um byte
- scanner.nextShort(); //para ler um short
- scanner.nextInt(); //para ler um inteiro
- scanner.nextLong(); //para ler um long
- scanner.nextFloat(); //para ler um float
- scanner.nextDouble(); //para ler um
- double scanner.nextBoolean(); //para ler um boolean
- scanner.next(); //para ler um uma palavra

Exercício #2

Crie uma calculadora que realiza a soma de dois números informados pelo usuário

Desafio #1

Modifique a calculadora para que posso realizar as outras operações matemáticas
Dica: Para comparar Strings utilizar o métodos equals()
“+” == “+” -> false
“+”.equals(“+”) -> true

Desafio #2

Modifique a calculadora para que possam ser realizadas mais do que um operação

OBRIGADO

FIAP