

C206 - POO

Arthur Openheimer

Estrutura de um código Java

- `public`
 - Modificador de acesso do método e da classe. Representa que podem ser acessados de qualquer outra classe no projeto
- `class`
 - Indica que Main é uma classe

```
public class Main {  
    Run | Debug  
    public static void main(String[] args) {  
  
    }  
}
```

- `static`
 - Define o método como estático, fazendo com que seja uma função da classe e não do objeto

Estrutura de um código Java

- (String[] args)
 - É o parâmetro do método main, um array de String chamado args. Serve caso o programa precise receber algum valor como argumento, comum quando o programa é iniciado por outro programa ou pelo terminal
- Main
 - Nome do método, main é o método principal e de entrada do programa. É ele que a JRE procura no início da execução do programa

```
public class Main {  
    Run | Debug  
    public static void main(String[] args) {  
  
    }  
}
```

Saída de dados

```
public class Main {  
    Run | Debug  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

Atalhos para ajudar no desenvolvimento:

- VSCode -> sysout
- IntelliJ -> sout

Tipos de dados

Tipos primitivos

Java é uma linguagem estaticamente tipada -> Variáveis sempre devem ser declaradas antes de usadas, com nome e tipo

Os tipos primitivos funcionam como já estamos acostumados em algumas outras linguagens

```
int numero = 10;  
boolean verdadeiro = false;  
double decimal = 10.5;  
char letra = 'A';
```

Tipos de dados

Tipos não primitivos

São tipos mais complexos que muitas vezes acompanham métodos que fornecem algumas vantagens

```
String nome = "ARTHUR";  
System.out.println(nome.toLowerCase());  
Integer idade = 20;  
System.out.println(idade.floatValue());
```

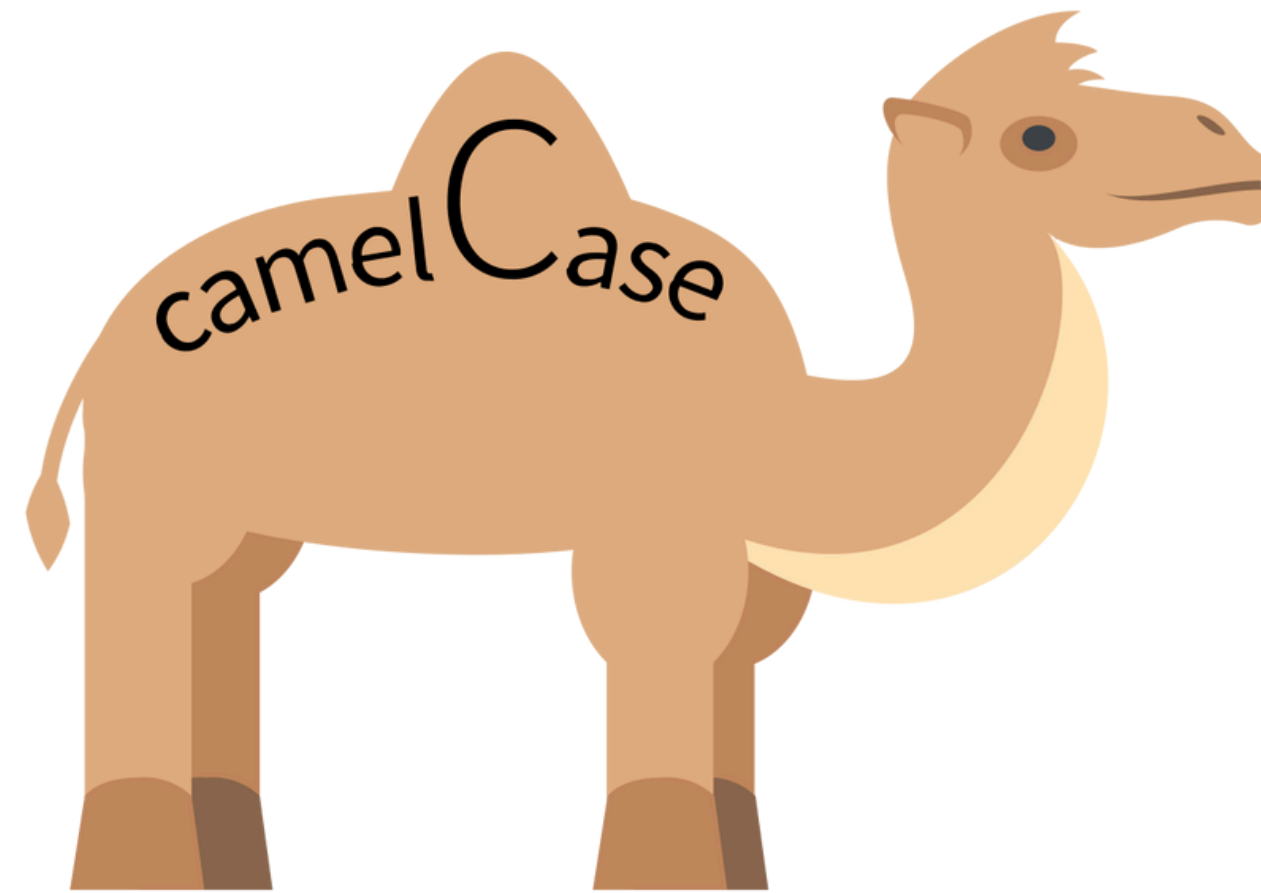


```
arthur  
20.0
```

Funções

```
public static float calcularMediaSemestre(float nota1, float nota2){  
    return (nota1 + nota2) / 2;  
}
```

Uma boa prática para nomear métodos é utilizar verbos, onde a primeira letra da primeira palavra é minúscula e a primeira letra das próximas palavras maiúsculas, esse padrão se chama camelCase



Exercícios

1. Crie um projeto java e faça um código com duas funções que são **chamadas na função main**:
 - a. Uma função que **calcule e imprime o fatorial de n** com recursividade, onde n é o parâmetro da função
 - b. Uma função que **calcule e imprime todos os números pares entre 0 e n**, onde n é o parâmetro da função

```
public static int calculaFatorial(int n){  
  
}  
  
public static void mostraPares(int n){  
  
}
```

```
public static void main(String[] args){  
  
    System.out.println(calculaFatorial(n:5));  
    mostraPares(n:10);  
}
```