

## Programação Orientada a Objetos

Your first program -

• Programando em Java:

↳ Criar um arquivo: `MeuPrograma.java`

↳ Compilar: `yavac MeuPrograma.java`

↳ Executar: `java MeuPrograma`

• Nome primeiro para criar o código (a mensagem que aparece no tela)

• No segundo passo será compilado o código que é traduzido para um código de máquina que o computador consegue entender e executar

• No terceiro passo será transferido o controle do computador do seu sistema para o seu programa

### Criando o programa em Java

↳ sequência de caracteres armazenados no arquivo `.java`

↳ O código pode ser escrito em qualquer editor de texto (Vscode etc...)

### Compilando um programa em Java

↳ o código escrito é feito para ser compreendido por humanos

↳ o código não é compreendido diretamente pelo computador

↳ Para o computador entender, ele é traduzido (compilado) para uma linguagem que o computador entende

Como funciona a compilação?

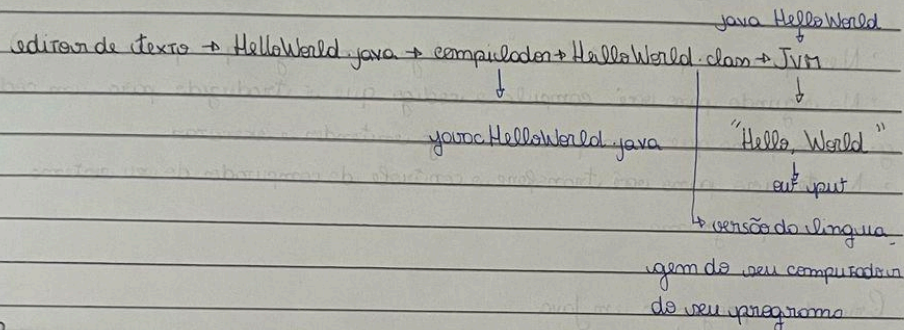
• Usa um compilador (`yavac`) que transforma o código Java em um código que o computador entende

• O computador cria um arquivo com o código em bytecode (linguagem intermediária que a máquina entende)



### Executando (rodando) o programa

- Depois de compilar o programa, você executa ele
- O programa assume o controle do computador
- Passa de um JVM para o código em byte code
  - ↳ todo o código em qualquer sistema



Programa:  $\rightarrow$  define uma classe chamada HelloWorld

public class HelloWorld { \* Programa em Java é organizado em classes

$\rightarrow$  método principal onde o programa começa a executar

public static void main (String[] args) {

$\rightarrow$  o programa começa nesse linha

System.out.println ("Hello, World");

$\rightarrow$  comando que exibe a mensagem no terminal

}

Pontos importantes sobre o programa Java

- Nome do arquivo deve ser o mesmo do classe público

HelloWorld.java

- Método é um bloco de código que realiza uma tarefa específica

$\rightarrow$  1ª linha  $\rightarrow$  nome do método e informações sobre como ele será utilizado

- Corpo do método é tudo entre chaves



- comentários // comentário de linha; /\* \*/ comentário de bloco
- `System.out.println("Hello, World!");`
  - ↳ comando de impressão, tudo entre aspas será mostrado no terminal

Erros de compilação (Bugs) → ao tentar compilar o código

Erros de execução → durante a execução

Erros lógicos → quando o resultado é errado, mas não há mensagem de erro

Como identificar e corrigir erros

- ler as mensagens de erro
- verificar se o nome do arquivo é o do classe
- Todas as chaves {} estão fechadas
- ; no final de cada comando

## Input e Output em Java

```
public class UseArgument {
    // armazenar os argumentos fornecidos
    public static void main (String[] args) {
        // na execução
        System.out.println("Hello, " + args[0] + "!");
    }
}
```

↳ 1º argumento fornecido pelo usuário

Como os argumentos são armazenados

• `args[0]` 1º argumento

• `args[1]` 2º argumento, e assim por diante

↳ não armazenados em arrays

## Diferença entre `print()` e `println()`


• `println()` → exibe uma mensagem e em seguida adiciona um quebra de linha

• `print()` → exibe a mensagem mas não adiciona um quebra de linha



Alice ← input string



 ← Block box

Hi, Alice! How are you? ← output string

° Sistema caixa preto mostra como um programa pode responder a informações de usuários

° O usuário insere o argumento  
para Use Argument Alice