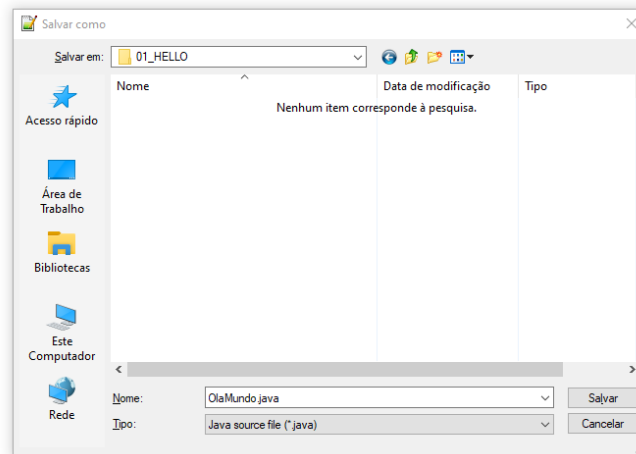


PROJECTJAVA: Ola Mundo

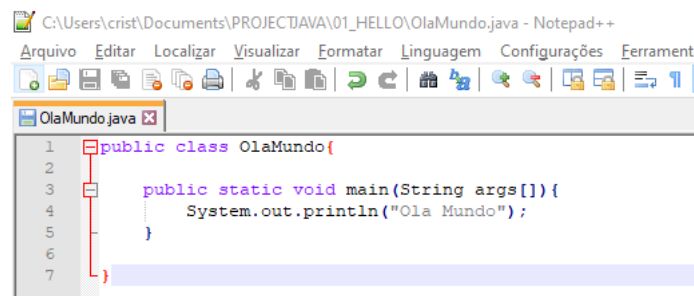
O menor programa em JAVA é simplesmente o envio de uma mensagem via terminal (prompt de comando/cmd). Não por ser pequeno, mas com este primeiro programa já é possível ver se o ambiente está configurado corretamente.

Aqui é utilizado o Notepad++, nada de uma IDE própria para programação JAVA nos passos básicos:



Nota: o nome do fonte JAVA DEVE iniciar com uma letra maiúscula e não deve ser precedida de números, símbolos, ou ainda palavras reservadas. Este arquivo é nomeado como OlaMundo.java. Este é o fonte! Após a compilação é gerado um arquivo .class que é o binário a ser interpretado pela máquina virtual JAVA (JVM).

O fonte é assim composto:



Na linha 1 a sentença public class OlaMundo{ indica o nome da classe, onde:

- public: indica a visibilidade da classe.
- class: indica que o objeto é uma classe, outro exemplo de objeto é o interface.
- OlaMundo: nome dado à classe.
- {: a chave indica o início do fonte que compõe a classe e deve ter encerrado com a chave } (linha 7).

Na linha 3 mais uma sentença: public static void main(String args[]){, onde:

- public: indica a visibilidade do método principal, analogamente a linguagem C, um programa JAVA possui uma função para executá-lo, mas aqui ela recebe o nome de método.
- static: indica que é um método estático.
- void: indica que não necessita de valor de retorno.
- main: nome do método.
- (String args[]): parâmetros que podem ser adicionados no momento da execução do programa.
- {: a chave indica o início do método e deve ter encerrado com a chave } (linha 5).

Leitura recomendada: <https://www.devmedia.com.br/trabalhando-com-metodos-em-java/25917>

Na linha 4 encontra-se a sentença: `System.out.println("Ola Mundo");`, onde:

- System: JAVA tudo se torna objeto, ou parte de uma classe, neste caso, o conjunto de classes padrão do JAVA é o `java.lang.System` que contém a maioria dos recursos necessários para o programa conversar com o sistema operacional.
- out: define que será utilizado a saída padrão, neste caso o monitor.
- `println()`: função de impressão de texto na tela do computador.

Observação:

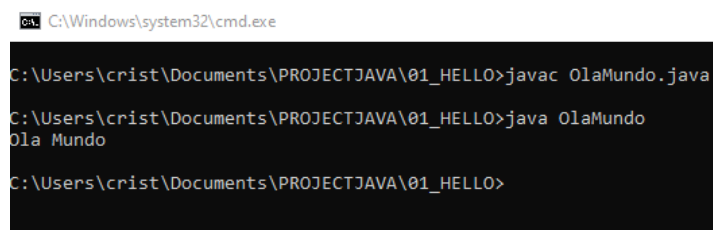
- Sintaxes iniciadas e terminadas com { } não necessitam do token ; para finalizar a linha.

Depois de escrito e salvo o fonte é hora de gerar os binários e rodar. Para tanto, acesse o diretório com o `OlaMundo.java` pelo terminal e rode os comandos abaixo para ver o resultado:

```
javac OlaMundo.java
```

```
java OlaMundo
```

Se tudo estiver correto irá observar a mensagem:



```

C:\Windows\system32\cmd.exe

C:\Users\crist\Documents\PROJECTJAVA\01_HELLO>javac OlaMundo.java
C:\Users\crist\Documents\PROJECTJAVA\01_HELLO>java OlaMundo
Ola Mundo
C:\Users\crist\Documents\PROJECTJAVA\01_HELLO>
  
```

Você pode estar se perguntando, e esse tal de `String args[]` dentro dos argumentos da função principal?

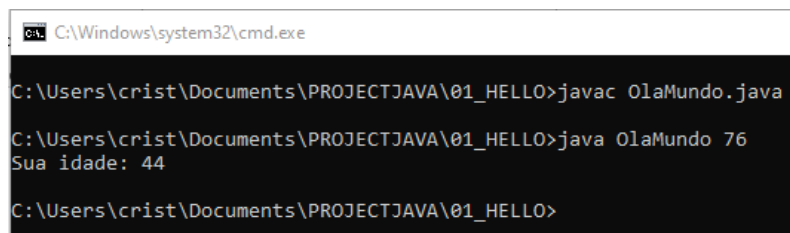
Ele é utilizado para passar alguma informação para o programa que pode ser a base de alguma frase, ou, algum cálculo. Modifique o programa `OlaMundo.java` de forma que ele receba o ano em que nasceu (por exemplo, se nasceu em 1976, passe somente os últimos dois dígitos) e o programa informará sua idade:

```
public class OlaMundo{  
    public static void main(String args[]){  
        System.out.println("Sua idade: " +  
                            (120 - Integer.parseInt(args[0])));  
    }  
}
```

A modificação ocorreu na linha de `System.out.println`, onde três coisas novas apareceram:

- `+`: o sinal de mais é utilizado para concatenar Strings em JAVA.
- `Integer`: tipo primário padrão pertencente a classe `java.lang.Integer` que possui um método que converte uma `String` para um valor inteiro (`parseInt()`);
- `args[0]`: indica que será utilizado o argumento 0, ou seja, o primeiro argumento para fazer o cálculo.

Salve, compile e rode o programa:



```
C:\Windows\system32\cmd.exe  
  
C:\Users\cris\Documents\PROJECTJAVA\01_HELLO>javac OlaMundo.java  
C:\Users\cris\Documents\PROJECTJAVA\01_HELLO>java OlaMundo 76  
Sua idade: 44  
C:\Users\cris\Documents\PROJECTJAVA\01_HELLO>
```