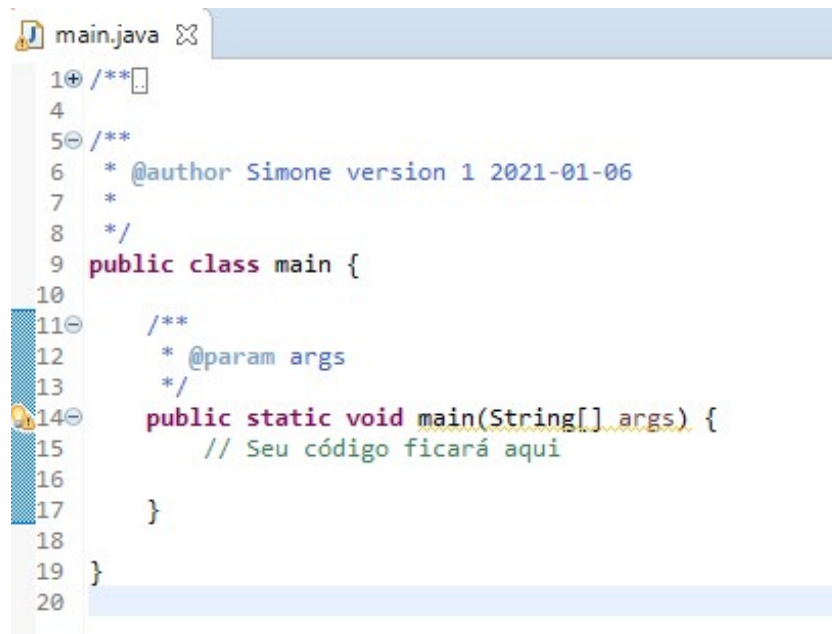


Lista 3

Criação de um projeto Java

Este exercício poderá ser feito em qualquer IDE. A diferença é que cada IDE fornece mais ou menos facilidades na criação do projeto. Algumas já criam uma árvore de pastas para a separação dos tipos de arquivos, em outros apenas uma pasta raiz onde a classe principal deverá ser criada. Para que os exercícios funcionem já deverá estar instalado no seu equipamento a JDK e JRE.

1. Usando a sua IDE crie um novo projeto Java. Dê um nome ao seu projeto e escolha um local para que seja salvo.
2. Se a IDE não criar uma classe principal, crie uma classe Java no seu projeto. Se a sua IDE criou uma pasta separada para os códigos-fontes (src), coloque a classe nesta pasta. Dê um nome para a classe principal e salve. Este arquivo deverá ter extensão `.java`.
3. A classe principal criada deverá ter o método principal `main()`. A execução do seu projeto iniciará deste método. A assinatura do método `main()` deverá ser feita da seguinte forma: `public static void main(String[] args)`. A sua classe principal deverá ficar muito similar ao que mostra a Figura 1:

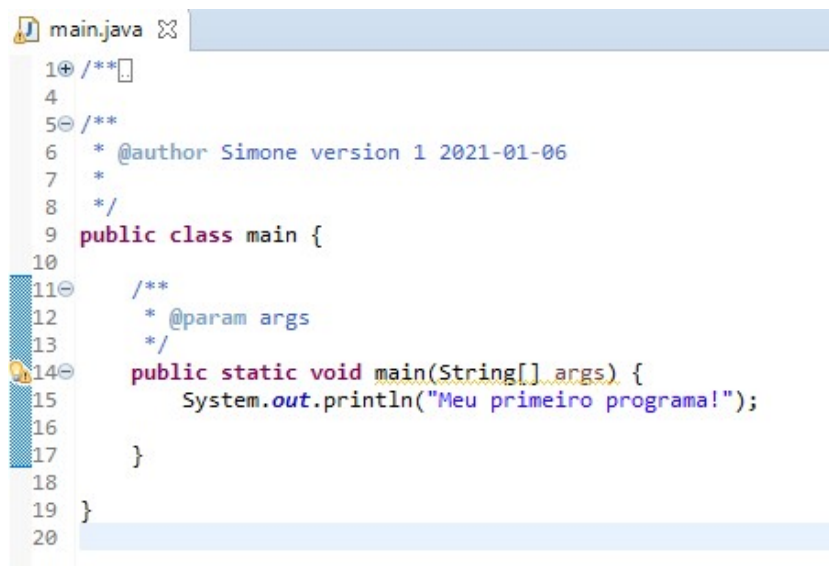


```
1  /**
4
5  /**
6   * @author Simone version 1 2021-01-06
7   *
8   */
9  public class main {
10
11     /**
12     * @param args
13     */
14     public static void main(String[] args) {
15         // Seu código ficará aqui
16
17     }
18
19 }
20
```

Figura 1:

4. Insira um comando para escrever na tela do console a mensagem "Meu primeiro programa". O comando `System.out.println()` escreve texto na tela do console.

A Figura 2 mostra como usar o comando. Execute a classe como uma aplicação Java.



```
main.java
1  /**
4
5  /**
6   * @author Simone version 1 2021-01-06
7   *
8   */
9  public class main {
10
11     /**
12     * @param args
13     */
14     public static void main(String[] args) {
15         System.out.println("Meu primeiro programa!");
16     }
17 }
18
19
20
```

Figura 2:

Observe que todos os blocos criados são feitos através de chaves, como na criação da classe:

```
public class main { }
```

e na criação do método main:

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

Todos os comandos são encerrados com ponto e vírgula.

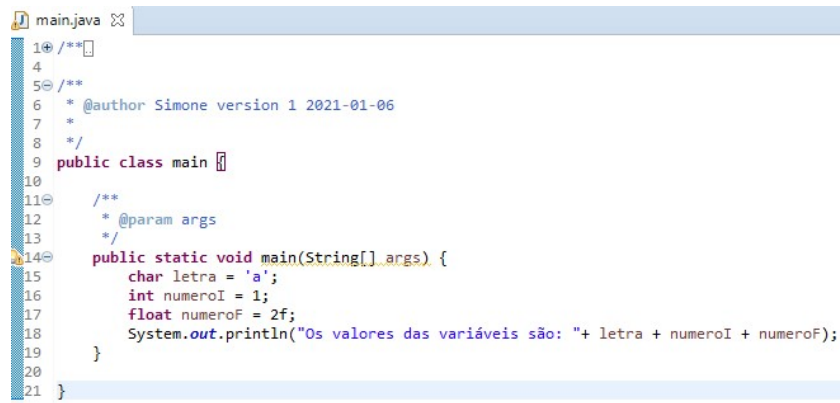
O comando `System.out.println()` imprime literalmente a string que aparece entre o par de aspas duplas. O método `println()` pertence ao objeto `System.out` que é construído na linguagem Java.

Quando é preciso imprimir o conteúdo de variáveis ou constantes, o comando fica mais complicado. Se a impressão é de apenas uma variável, não se usa as aspas duplas. Se a impressão é de string com variáveis, deve ser feita a concatenação dos dois conteúdos usando o operador “+” para constituir um único parâmetro de string.

A criação de variáveis é feita no mesmo estilo da linguagem C, com a declaração explícita antes do uso, e inserindo primeiro o tipo e após o nome da variável.

5. Altere o código da questão anterior, crie variáveis e atribua valores a elas. Mostre o conteúdo das variáveis criadas acompanhadas de algum texto. A Figura 3 mostra um exemplo de resultado. Execute o seu código e veja o resultado.

No exemplo mostrado na figura, o valor das variáveis ficou colado um no outro, como resolver?



```
14 /**
4
5 /**
6  * @author Simone version 1 2021-01-06
7  *
8  */
9  public class main {
10
11      /**
12       * @param args
13       */
14      public static void main(String[] args) {
15          char letra = 'a';
16          int numeroI = 1;
17          float numeroF = 2f;
18          System.out.println("Os valores das variáveis são: " + letra + numeroI + numeroF);
19      }
20
21 }
```

Figura 3:

6. Usando as estruturas de controle de fluxo, faça alguns exemplos para praticar o condicional e as estruturas de repetição.