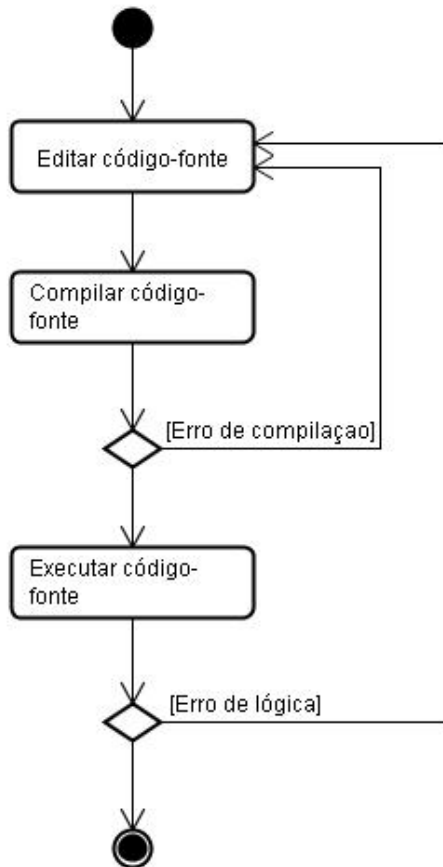


Simulado de Linguagem de Programação Java

1. Descreva o processo de criação de “programas” em Java, desde a criação do código-fonte até a obtenção do resultado esperado, por meio de sua execução. Cite as instruções e/ou comandos usados durante o processo.

- Passo 1: criar e/ou editar o código-fonte usando um editor de textos ASCII
- Passo 2: compilar o código-fonte com `javac <nome_da_classe>.java`
- Passo 3: executar a classe com `Java <nome_da_classe>`



2. Descreva sucintamente em que situações podem ocorrer os erro e exceções listados abaixo e quais as ações necessárias para corrigi-los.

a. **NoSuchMethodError: main**

Causa: tentativa de executar um classe sem o método main.

Solução: verificar se o método existe e se a sua assinatura está correta `public static void main(String a[])`.

b. **NoClassDefFoundError: principal**

Causa: tentativa de executar uma classe que não existe.

Solução: verificar se o nome da classe este correto em sua declaração.

c. **ArrayIndexOutOfBoundsException: 2**

Causa: tentativa de acessar o elemento de índice 2 de um array.

Solução: verificar erro de lógica e se todos os argumentos da linha de comando, esperados pelo programa, foram de fato passados.

d. **incompatible types: found <tipo1> required <tipo1>**

Causa: tentativa de usar um tipo de dado <tipo1> onde era esperado <tipo1>

Solução: corrigir erro de lógica, definindo o tipo adequado e compatível

e. **NumberFormatException: For input string "xpto"**

Causa: tentativa de gerar um tipo numérico a partir de um String, sendo que o String não pode ser convertido em número.

Solução: tratar o erro com try/catch ou assegurar de outra forma que o String que se deseja converter contenha apenas dígitos.

f. **cannot find symbol: <qual> <onde>**

Causa: alguma variável ou método <qual> usado não está definido em <onde>.

Solução: definir o que não está definido ou corrigir erro de digitação.

g. **javac: file not found: Teste.java**

Causa: tentativa de compilar um código-fonte (Teste.java) que não existe na pasta atual.

Solução: verificar a existência do arquivo, alterando sua localização e/ou nome.

3. Escrever uma classe “executável” em Java que recebe, pela linha de comando, um número inteiro e exibe a quantidade de divisores exatos do número recebido.

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

        int qtdDiv = 0, n;
        n = Integer.parseInt(args[0]);
        for (int i = 1; i <= n; i++) {
            if (n % i == 0) {
                qtdDiv++;
            }
        }
        System.out.println(n + " tem " + qtdDiv + " divisores!!!");
    }
}
```

4. Escrever uma classe “executável” em Java que recebe, pela linha de comando, um número inteiro e exibe a soma dos divisores exatos do número recebido.

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

        int soma = 0, n;
        n = Integer.parseInt(args[0]);
        for (int i = 1; i <= n; i++) {
            if (n % i == 0) {
                soma = soma + i;
            }
        }
        System.out.println("A soma dos divisores exatos de " + n + " é = " + soma);
    }
}
```

5. Escrever uma classe “executável” em Java que lê um número não determinado de valores, todos inteiros e positivos, pela linha de comando. Para cada número, informar se o número é par ou ímpar.

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

        if (args.length == 0) {
            System.out.println("Erro na quantidade de argumentos!!!");
        } else {
            for (int i = 0; i < args.length; i++) {
                int n = Integer.parseInt(args[i]);
                if (n % 2 == 0) {
                    System.out.println(n + " é par!!!");
                } else {
                    System.out.println(n + " é ímpar!!!");
                }
            }
        }
    }
}
```

6. Escrever uma classe “executável” em Java que lê uma quantidade não determinada de valores inteiros e positivos, pela linha de comando, e exibe, para cada número, a soma dos inteiros de um até o número lido, inclusive.

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

        int soma, n;

        for (int i = 0; i < args.length; i++) {
            soma = 0;
            n = Integer.parseInt(args[i]);
            for (int a = 1; a <= n; a++) {
                soma = soma + a;
            }
            System.out.println(
                "A soma dos numeros inteiros ate " + n + " é = " + soma);
        }
    }
}
```

7. Escrever uma classe “executável” em Java que recebe, pela linha de comando, um número inteiro e exibe o fatorial do número recebido.

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

        int soma, n;
        n = Integer.parseInt(args[0]);
        int fat = 1;
        for (int i = n; i > 1; i--) {
            fat = fat * i;
        }
        System.out.println("Fatorial de " + n + " é = " + fat);
    }
}
```

8. Escrever uma classe “executável” em Java que lê um número não determinado de valores, todos inteiros e positivos, pela linha de comando. Para cada número, informar se o número é primo ou não.

```
class Primo {
    public static void main(String[] args) {
        for(int a = 0; a < args.length; a++){
            int n = Integer.parseInt(args[a]);
            boolean primo = true;
            for (int i = 2; i < n; i++) {
                if (n % i == 0) {
                    primo = false;
                    break;
                }
            }
            if (primo == true){
                System.out.println(n + " é primo");
            }else{
                System.out.println(n + " não é primo");
            }
        }
    }
}
```

9. Escrever uma classe “executável” em Java que lê dois valores, inteiros e positivos, e exibe o Máximo Divisor Comum (MDC) dos números.

```
class Mdc {
    public static void main(String[] a) {

        int a = Integer.parseInt(a[0]);
        int b = Integer.parseInt(a[1]);

        int menor = a;
        if (b < a) {
            menor = b;
        }
        int mdc = 1;
        for (int i = 1; i <= menor; i++) {
            if (a % i == 0 && b % i == 0) {
                mdc = i;
            }
        }
        System.out.println("Máximo divisor comum: " + mdc);
    }
}
```

10. Escrever uma classe “executável” em Java que lê dois valores, inteiros e positivos, e exibe Mínimo Múltiplo Comum (MMC) dos números.

```
class Mmc {
    public static void main(String[] a) {

        int a = Integer.parseInt(a[0]);
        int b = Integer.parseInt(a[1]);

        int maior = a;
        if (b > a) {
            maior = b;
        }
        int mdc = maior;
        for (int i = maior; i <= a * b; i++) {
            if (i % a == 0 && i % b == 0) {
                mmc = i;
                break;
            }
        }
        System.out.println("Mínimo Multiplo Comum: " + mmc);
    }
}
```