LÓGICA DE PROGRAMAÇÃO

“Subalgoritmos são trechos de algoritmos que efetuam um ou mais cálculos determinados. Ao invés de escrever um algoritmo grande, escrevem-se vários algoritmos menores, os quais, não isoladamente mas em conjunto, resolvem o problema proposto.

SUBALGORITMOS

- Promovem o reaproveito de código de fato, uma vez que podem ser utilizados em várias partes do algoritmo;
- Nesse caso, ao invés de escrever um conjunto de instruções N vezes, escreve-se apenas uma vez e sempre que for preciso em outras partes do algoritmo, invoca-se o subalgoritmo;
- Se bem modelados os subalgoritmos, é possível tornar o código muito mais limpo e simples de entender;

SUBALGORITIMOS

- De forma simplificada, é possível observar as seguintes vantagens:
  - Reduzem o tamanho do algoritmo;
  - Simplificam a compreensão e a visualização do algoritmo;
  - Podem ser chamados em qualquer ponto do algoritmo após a sua declaração;
  - Podem ser funções ou procedimentos, onde o primeiro retorna um valor enquanto o segunda não possui retorno algum;

SUBALGORITMOS - FUNÇÃO

- Uma função, é um mecanismo de encapsulamento de código que permite retornar um valor ou uma informação;
- A chamada de uma função é realizada por meio da citação do seu nome seguindo opcionalmente, de seus argumentos (parâmetros);
- As funções podem ser predefinidas pela linguagem ou criadas pelo desenvolvedor de acordo com a necessidade;
- A linguagem Java possui milhares de funções predefinidas que podem ser utilizadas pelo programador;

SUBALGORITMOS - FUNÇÕES JAVA I

Função	Descrição
Math.pow(a,b)	Retorna o valor de ‘a’ elevado a ‘b’
Math.sqrt(a)	Retorna o valor da raiz quadrada de ‘a’
Math.min(a, b)	Retorna o menor valor entre ‘a’ e ‘b’
Math.max(a, b)	Retorna o maior valor entre ‘a’ e ‘b’
Math.ceil(a)	Retorna o valor de ‘a’ arredondado ‘para cima’
Math.floor(a)	Retorna o valor de ‘a’ arredondado ‘para baixo’
Math.random()	Retorna um valor positivo aleatório entre 0.0 e 1.0 (inclusos)

Fonte: https://docs.oracle.com/javase/6/docs/api/java/lang/Math.html

SUBALGORITMOS - FUNÇÕES JAVA II

Função	Descrição
String.charAt(<indice>)	Retorna um char em uma posição da String
String.endsWith(<outra string>)	Retorna um booleano caso a String termine com o parâmetro informado
String.startsWith(<outra string>) String.startsWith(<outra string>, <indice>)	Retorna um booleano caso a String comece com o parâmetro informado.
String.toCharArray()	Retorna um vetor de char[] a partir da String
String.split()	Por meio de regex, converte uma String em um vetor de substring's retornando o mesmo
String.toLowerCase()	Retorna uma String com todos os caracteres em minúsculo
String.toUpperCase()	Retorna uma String com todos os caracteres em maiúsculo

Fonte: https://docs.oracle.com/javase/6/docs/api/java/lang/String.html

SUBALGORITIMOS

```java
public class FuncoesMatematicas {

    static double resultadoPow;
    static double resultadoSqrt;
    static double resultadoMin;
    static double resultadoMax;
    static double resultadoCeil;
    static double resultadoFloor;
    static double resultadoRandom;

    public static void main(String[] args) {

        resultadoPow = Math.pow(2, 3);//8.0
        resultadoSqrt = Math.sqrt(2);//1.4142135623730951
        resultadoMin = Math.min(5, 2);//2.0
        resultadoMax = Math.max(8, 6);//8.0
        resultadoCeil = Math.ceil(14.2);//15
        resultadoFloor = Math.floor(16.5);//16
        resultadoRandom = Math.random();//0.37728049573365774

    }

}
```

```java
public class FuncoesLiterais {

    static String nomes;
    static char retornoCharAt;
    static boolean retornoEndsWith;
    static boolean retornoStartsWith;
    static char[] retornoToCharArray;
    static String[] retornoSplit;
    static String retornoToLowerCase;
    static String retornoToUpperCase;

    public static void main(String[] args) {

        nomes = "José;Glauco;Marcos";
        retornoCharAt = nomes.charAt(0);//'J'
        retornoEndsWith = nomes.endsWith("os");//true
        retornoStartsWith = nomes.startsWith("Mar");//false
        //['J'|'o'|'s'|'é'|';'|'G'|'l'|'a'|'u'|'c'|'o'|';'|'M'|'a'|'r'|'c'|'o'|'s']
        retornoToCharArray = nomes.toCharArray();
        //["José"|"Glauco"|"Marcos"]
        retornoSplit = nomes.split(";");
        retornoToLowerCase = nomes.toLowerCase();//"josé;glauco;marcos"
        retornoToUpperCase = nomes.toUpperCase();//"JOSÉ;GLAUCO;MARCOS"

    }

}
```

SUBALGORITMOS - FUNÇÕES

- A criação de uma função em pseudocódigo deve ocorrer juntamente com a declaração das variáveis no começo do algoritmo;
- Uma função deverá sempre retornar um valor para o código que a invocou, e esse valor deve representar um dos tipos de variáveis aprendidos no curso;
- Os parâmetros de uma função são como variáveis passadas do algoritmo principal para essa mesma função, e poderão ser utilizados normalmente dentro desse subalgoritmo;

FUNÇÕES EM PSEUDOCÓDIGO

```
Algoritmo "<nome do algoritmo>"
var
   <declaração de variáveis globais>
<definição da função>
inicio
   < lista de comandos>
fimalgoritmo
```

```
funcao <identificador> ([var]<parâmetros>) <tipo de retorno>
var
<declaração de variáveis locais>
inicio
<lista de comandos>
retorne <variável de retorno>
fimfuncao
```

FUNÇÕES EM PSEUDOCÓDIGO

```
1.    ALGORITMO "Funções Personalizadas"
2.    var
3.    Valor_1,Valor_2, soma: real
4.
5.    FUNCAO FSoma(Recebe_valor1, Recebe_valor2: Real):Real
6.    var
7.    total : real
8.    Inicio
9.    total<-Recebe_valor1+Recebe_valor2
10.   retorne total
11.   fimfuncao
12.
13.   INICIO
14.   Escreva ("Valor_1 : ")
15.   LEIA (Valor_1)
16.
17.   Escreva ("Valor_2 : ")
18.   LEIA (Valor_2)
19.   soma<-FSoma(Valor_1,Valor_2)
20.
21.   ESCREVA ("Soma das vaiáveis é ", soma)
22.   FIMALGORITMO
```

FUNÇÕES EM JAVA

- Funções em Java são bem similares as funções em pseudocódigo;
- Nesse caso o que muda são as posições dos elementos constituintes;
- A ordem de declaração no algoritmo principal nesse momento, será mantida do mesmo jeito que em pseudocódigo;
- Isso se deve para manter o conteúdo programático alinhado com a nova proposta abordada na unidade curricular;

FUNÇÕES EM JAVA

```java
import javax.swing.JOptionPane;

public class FuncoesConstruidas {

    static double resultadoDaSoma;
    static double primeiroNumero;
    static double segundoNumero;

    static double somar(double numero1, double numero2) {
        double resultado;
        resultado = numero1 + numero2;
        return resultado;
    }

    public static void main(String[] args) {
        primeiroNumero = Double.parseDouble(
                JOptionPane.showInputDialog("Digite o primeiro número:"));
        segundoNumero = Double.parseDouble(
                JOptionPane.showInputDialog("Digite o segundo número:"));
        resultadoDaSoma = somar(primeiroNumero, segundoNumero);
        JOptionPane.showMessageDialog(null, "O resultado da soma é: " + resultadoDaSoma);
    }

}
```

FUNÇÕES: PSEUDOCÓDIGO X JAVA

```
1.    ALGORITMO "Funções Personalizadas"
2.    var
3.    Valor_1,Valor_2, soma: real
4.
5.    FUNCAO FSoma(Recebe_valor1, Recebe_valor2: Real):Real
6.    var
7.    total : real
8.    Inicio
9.    total<-Recebe_valor1+Recebe_valor2
10.   retorne total
11.   fimfuncao
12.
13.   INICIO
14.   Escreva ("Valor_1 : ")
15.   LEIA (Valor_1)
16.
17.   Escreva ("Valor_2 : ")
18.   LEIA (Valor_2)
19.   soma<-FSoma(Valor_1,Valor_2)
20.
21.   ESCREVA ("Soma das vaiáveis é ", soma)
22.   FIMALGORITMO
```

```java
import javax.swing.JOptionPane;

public class FuncoesConstruidas {

    static double resultadoDaSoma;
    static double primeiroNumero;
    static double segundoNumero;

    static double somar(double numero1, double numero2) {
        double resultado;
        resultado = numero1 + numero2;
        return resultado;
    }

    public static void main(String[] args) {
        primeiroNumero = Double.parseDouble(
                JOptionPane.showInputDialog("Digite o primeiro número:"));
        segundoNumero = Double.parseDouble(
                JOptionPane.showInputDialog("Digite o segundo número:"));
        resultadoDaSoma = somar(primeiroNumero, segundoNumero);
        JOptionPane.showMessageDialog(null, "O resultado da soma é: " + resultadoDaSoma);
    }

}
```

MÃO NA MASSA

Caso 1

Construa um algoritmo que realize uma das quatro operações matemáticas sobre dois números decimais positivos. O usuário deve fazer uso de um menu para selecionar qual operação deseja realizar, e após informar os números, o resultado deve ser apresentado. O algoritmo deve ser modularizado em 4 funções além do algoritmo principal.

Caso 2

Crie um algoritmo que a partir do nome completo informado pelo usuário e mostre apenas o primeiro nome. O algoritmo deve ser modularizado em uma função responsável por extrair o primeiro nome e devolver para o algoritmo principal.

MÃO NA MASSA

Caso 3

Construa um algoritmo que realize a leitura de um vetor de nomes de 5 posições e apresente ao final quantos nomes começam com um vogal. O algoritmo deve ser modularizado para que a contagem de nomes começando com vogal seja retornada ao algoritmo principal por meio de uma função.

Caso 4

Desenvolva um algoritmo que realize a leitura de salário e nome de 5 funcionários. Os nomes e salários devem ser armazenados nos devidos vetores respeitando o tipo de cada dado. Ao final um terceiro vetor de salários reajustados deve ser apresentado ao usuário. O algoritmo deve ser modularizado para que o reajuste seja realizado por uma função, retornando o resultado ao final ao algoritmo principal.