

Algoritmos (Aula 2)

Sequência lógica e ordenada e finita de passos para resolver um problema.

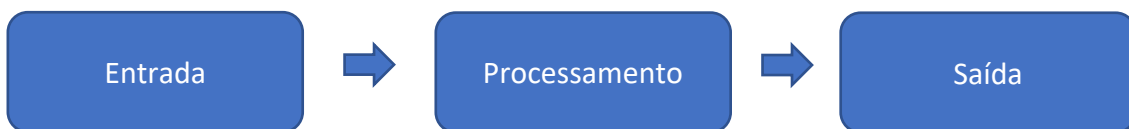
Para resolver um problema:

- Entender o problema
Entender **O QUÊ**
- 1. **COMO** resolver? (algoritmo / lógica de raciocínio)
- 2. **CRIAR** a solução (programa)

PROGRAMAR

- | | |
|--|------------|
| 1. Escrever o algoritmo em uma linguagem de programação (alto nível de abstração) | EDIÇÃO |
| 2. Transformar o programa em um código executável | COMPILAÇÃO |
| 3. Executar :: execução | EXECUÇÃO |

Para programar iniciamos com uma sequência simples:



Toda informação a ser utilizada em um programa precisa estar na memória (RAM):

- Reservar espaços na memória chamados **variáveis**.

Toda variável tem:

- um identificador
Não tem acentos, cedilha, espaço ou caractere especial
- um tipo
Define o tipo da informação que será armazenada

Os identificadores também tem algumas regras e podem seguir um estilo:

1. Nomes de variáveis iniciam por letra ou _ (underscore)
2. O estilo para variáveis
em java é o **camel Case**: nomeDaVariavel;
em linguagem C (**snake case**): nome_da_variavel;
em JavaScript (**kebab case**): nome-da-variavel;
em Pascal Case (**pascal case**): NomeDaVariavel.

Outros identificadores:

NomesDeClasse
nomesDeMetodo (conjunto de parâmetro)

Tipos de Variáveis Primitivas no Java

int	Valores inteiros, ocupa 4 bytes: -2^{31} a $2^{31}-1$
double	valores com decimais
char	para caracteres
boolean	para verdadeiro ou falso

Atribuição e Operações Aritméticas Simples

= atribui o valor de uma expressão a uma variável, sempre da direita para a esquerda.

Operações Aritméticas Simples

+	Soma
-	Subtração
*	Multiplicação
/	Divisão
%	Resto da divisão (módulo)

Declaração e Inicialização de Variáveis

Toda vez que vamos utilizar variáveis, precisamos declarar essas variáveis, ou seja, avisar o compilador para que reserve esses espaços adequadamente na memória.

Exemplos de declaração:

```
int a = 4, valorInicial;  
double taxaJuro;  
char opção = 'x';  
boolean continuarComprando;
```

Deve-se finalizar toda instrução com ponto-vírgula “;”

Expressões

```
int a = 21, b = 4, c;  
c = a + b; //c = 25  
int d = a * b; //d = 84  
int e = a / b; //e = 5  
int f = a % b; //f = 1  
double x = a / b; //x = 5.0  
double j = 21.0, k = 4.0;  
double y = j / k; //y = 5.25  
int m = j / b;  
double p = j / b;
```

Exibição de Dados

Para exibir uma saída no sistema, utilizamos o método **println**

Esse método é responsabilidade de System.out

Para exibir uma mensagem, utilizamos:

System.out.println("A mensagem entre aspas")

Para exibir uma mensagem e valores de variáveis, utilizamos:

System.out.println ("mensagem" + var1 + "outra mensagem"+ var2 + ...)

Código da Aula

```
public class Main {  
    public static void main(String args[]) {  
  
        int a=21, b=24;  
        double j=21.0, k=4.0;  
        //int m = j/b; erro de compilação  
        double p = j/b;  
        System.out.println(j + " / " + b + " = " + p);  
  
        char letra = 'p';  
        System.out.println("Minha letra = " + letra);  
  
        boolean sabadou = true;  
        System.out.println("Sabadou??? " + sabadou);  
    }  
}
```