Entradas e Saídas

Lógica de Programação I

Entradas e Saídas

É comum querermos apresentar alguma mensagem e/ou resultado para usuário durante a criação de nossos programas, podemos realizar isso através do terminal do computador.

Saídas

Como já vimos até aqui, podemos realizar isso utilizando as saídas em Java que podem ser feitas através da instrução:

```
System.out.println(expression);
```

É desta forma que escrevemos nosso primeiro programa "Hello World":

```
System.out.println("Hello World");

// resultado esperado > Hello World
```

Neste exemplo, temos que a expressão a ser escrita é uma simples string. Podemos variar este mesmo comportamento declarando essa string em uma variável e informando a variável como expressão:

```
String greeting = "Hello World";

System.out.println(greeting);

// resultado esperado > Hello World
```

Saídas

Ainda podemos incrementar mais e usar os operadores que aprendemos para combinar (concatenar) strings. Vamos declarar uma variável com o nome de alguém e emitir uma saudação personalizada:

```
String name = "Doctor Who";

System.out.println("Hello, " + name);

// resultado esperado > Hello, Doctor Who
```

Mas se ainda quisermos mais liberdade para formatar nossa saída, podemos optar pelo método printf. Sua sintaxe é:

```
System.out.printf(format,
        arguments);
```

Em format nós escrevemos a expressão que será escrita em console usando a sintaxe *Format String Syntax*. Essa sintaxe permite o uso de marcadores de argumentos, ou seja, cada marcador define uma variável que será substituída pelo valor contido nos arguments.

O segundo parâmetro, arguments, contém uma sequência de valores, separados por vírgulas, correspondente a cada marcador definido no primeiro parâmetro (format).

Saídas

Vamos analisar o seguinte exemplo:

```
String name = "Doctor Who";

System.out.printf("Hello, %s.", name);

// resultado esperado > Hello, Doctor Who.
```

Saídas

Existe um marcador específico para cada tipo de dados:

- *s* – formats strings
- *d* – formats decimal integers
- *f* – formats the floating-point numbers
- *t* – formats date/time values

Entradas

Entradas e Saídas

Agora que já aprendemos como escrever valores em console, vamos aprender um pouco sobre como ler esses valores. Isso nos levará a outro nível de poder de programação com a linguagem Java.

Até aqui, todos os valores de variáveis com os quais trabalhamos foram definidos por nós no próprio código-fonte. Isso "amarrou" nossos programas a sempre funcionarem da mesma forma a cada execução.

Agora, no entanto, poderemos ler valores inseridos pelos usuários e deixar nosso programa mais dinâmico.

Entradas

String[] args

A forma mais simples de ler entradas do usuário do programa é através do array `args` contido na assinatura do método `main`. Vamos relembrar a sintaxe deste método:

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

O array `args` é preenchido com os argumentos informados ao nosso programa **durante a inicialização** dele. Isto é, quando vamos executar nosso programa, podemos definir valores, separados por espaços, como argumentos.

Entradas

String[] args

Por exemplo, o programa a seguir deve receber o nome do usuário como argumento e exibir uma saudação personalizada.

```
public class Aplicacao {

    public static void main(String[] args) {

        System.out.printf("Hello, %s.", args[0]);

   }

}
```

Entradas

java.util.Scanner

A abordagem usando o array `args` serve apenas para cenários simples, onde todos os argumentos do programa serão informados logo em sua inicialização, mas não nos permite realizar algo interativo, onde vamos recebendo "*inputs*" passo-a-passo do usuário ou definindo o comportamento da aplicação de forma dinâmica.

Exercícios

Exercício 1: Identificar Número Primo

Peça ao usuário para inserir um número inteiro usando **Scanner**. Utilize o operador ternário para determinar se o número inserido é primo ou não, sem usar estruturas de controle de fluxo.

Exercício 2: Verificar Dia da Semana

Peça ao usuário para inserir um número de 1 a 7 usando Scanner, onde 1 representa "Domingo" e 7 representa "Sábado".

Utilize o operador ternário para exibir o dia da semana correspondente.

Exercício 3: Verificar Signo

Peça ao usuário para inserir um mês e dia de nascimento. Utilize `printf` para exibir o signo correspondente e ternário para as verificações.

- **Mu de Áries:** 21 de março a 20 de abril.
- **Aldebaran de Touro:** 21 de abril a 20 de maio.
- **Saga de Gêmeos:** 21 de maio a 20 de junho.
- **Máscara da Morte de Câncer:** 21 de junho a 22 de julho.
- **Aiolia de Leão:** 23 de julho a 22 de agosto.
- **Shaka de Virgem:** 23 de agosto a 22 de setembro.
- **Dohko de Libra:** 23 de setembro a 22 de outubro.
- **Milo de Escorpião:** 23 de outubro a 21 de novembro.
- **Aiolos de Sagitário:** 22 de novembro a 21 de dezembro
- **Shura de Capricórnio:** 22 de dezembro e 20 de janeiro
- **Camus de Aquário:** 21 de janeiro a 18 de fevereiro
- **Afrodite de Peixes:** 19 de fevereiro a 20 de março

Exercício 4: Identificar o Mes

Peça ao usuário para inserir um número de 1 a 12. Utilize `print` para exibir o nome do mês correspondente e ternário para as verificações.

Exercício 5: Exibir o Maior

Peça ao usuário para inserir três números inteiros. Utilize `println` para mostrar o maior número inserido.

Obrigada