sãojudas universidade	FTCE Faculdade de Tecnologia e Ciências Exatas

USJT – 2021/1 –Programação de Soluções Computacionais

Professores: Bonato e Keity

Aula: 02 – Algoritmos Básicos: variáveis e entrada e saída de dados

1 Introdução

Neste material iremos estudar os algoritmos de sequência simples, ou seja, aqueles que envolvem um conjunto de entradas, processamento e um conjunto de saídas. Faremos a implementação desses algoritmos utilizando a linguagem Java, para tanto, veremos também declaração de variáveis e métodos de entrada e saída.

2 IDEs para facilitar a vida do desenvolvedor

Um IDE é um software que permite que se execute as 3 fases do desenvolvimento de programas: editar, compilar e executar, daí o nome *Integrated Development Environment* – IDE (Ambiente Integrado de Desenvolvimento). Existem vários IDEs disponíveis no mercado, vamos começar com uma simples, para pegar o jeito: o JGrasp.

2.1 Meu primeiro programa em Java. Vamos abrir o IDE escolhido e digitar o programa ilustrado na Figura 2.1, ou seja, vamos editar o programa e salvá-lo como `HelloWorld.java`. Essa é a primeira regra de ouro: o nome do arquivo Java deve ser o nome da única classe pública contida no arquivo.

Figura 2.1

```
public class HelloWorld {

    public static void main(String[] args) {
        System.out.println("Hello World!!!");
    }

}
```

Agora, conforme visto na Aula 01, Seção 5.1, o código de um programa de computador escrito na linguagem Java é compilado para uma forma intermediária de código denominada bytecode, que é interpretada pelas Máquinas Virtuais Java. Para isso, utilizamos o comando `javac` ou acionamos o item do IDE que o invoca. A sintaxe do comando é descrita na Figura 2.2.

Figura 2.2

```
javac arquivo.java
```

Pronto, agora o nosso bytecode está pronto para ser "executado" pela JVM, o que pode ser feito chamando-se o comando `java` ou acionamos o item do IDE que o invoca. A sintaxe desse comando é descrita na Figura 2.3.

Figura 2.3

```
java arquivo
```

2.2 O arquivo HelloWorld.java. Quase tudo no Java é classe. O exemplo mostra que a classe `HelloWorld` tem um método `main`, que vai ter sempre a mesma sintaxe. A função `println` exibe uma mensagem em tela de comando. Vamos praticar algumas variações, utilizando a classe JOptionPane, por exemplo.

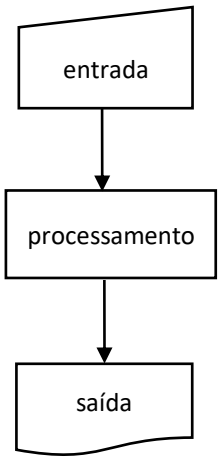
Observações importantes e boas práticas de programação:

1. blocos são delimitados por `{}`;
2. uma boa prática de programação é endentar sempre programas;
3. classes têm nomes iniciando por letra em caixa alta; variáveis e métodos têm nome iniciando por letras em caixa baixa; métodos sempre têm parênteses para argumentos, mesmo que fiquem vazios.
4. para quaisquer nomes, respeitamos o CamelCase[1]

3 Algoritmos Básicos

Um algoritmo básico ou de sequência simples tem a seguinte estrutura: entrada – processamento – saída. A Figura 3.1 ilustra um fluxograma para essa estrutura.

Figura 3.1



Um exemplo simples: somar 2 números escolhidos pelo usuário e exibir o resultado.

[1] CamelCase é a denominação em inglês para a prática de escrever as palavras compostas ou frases, sem espaços entre elas, e cada palavra é iniciada com letra maiúscula.

Entrada: informações que são fornecidas ao programa para que ele seja executado. Nesse primeiro exemplo, são os 2 números escolhidos pelo usuário.

➔ ler 2 números e armazená-los nas variáveis:
`primeiroValor` e `segundoValor`

Processamento: é a função do programa. No exemplo é realizar a soma, também armazenando o resultado em uma variável:

➔ `resultado = primeiroValor + segundoValor`

Saída: exibição do resultado obtido. Para o exemplo, exibir o valor da soma.

➔ `exibir resultado`

3.1 Armazenamento dos valores na memória: as variáveis

Uma variável é uma área reservada na memória RAM, identificada por um nome, que pode armazenar valores de um determinado tipo. Um tipo de dado define um conjunto de valores e um conjunto de operações válidos.

No Java temos vários tipos chamados primitivos que permitem trabalhar com vários tipos de informação, vamos ver alguns deles:

- ➔ `int`: é o tipo de dado capaz de armazenar 32 bits, ou seja, de representar um número inteiro qualquer entre -2.147.483.648 e 2.147.483.647.
- ➔ `double`: permite armazenar valores de ponto flutuante IEEE 754 de 64 bits e dupla precisão. Essa é a opção padrão para valores decimais.
- ➔ `boolean`: armazena um único byte de informação, que pode ser representado pelas palavras **false** (falso) ou **true** (verdadeiro).

A Figura 3.1.1 mostra o esqueleto da classe `SomaDoisNumeros`, com a definição do bloco do método `main` e a declaração das 3 variáveis, do tipo `double`. Perceba a endentação do código.

Figura 3.1.1

```
public class SomaDoisNumeros {
    public static void main (String[] args) {
        double primeiroValor, segundoValor, resultado;
    }
}
```

3.2 Entrada de Dados

O Java provê várias classes que podem realizar a entrada de dados. Vamos iniciar pela classe `JOptioPane`, que fornece, entre outros, métodos para entrada e saída. Essa classe está no pacote `javax.swing`, portanto, devemos importar para podermos usar.

O método para entrada de dados é o `showInputDialog`. A Figura 3.2.1 mostra agora a classe `SomaDoisNumeros`, com a linha de importação da classe `JOptioPane` e a leitura (entrada) dos valores que o usuário digita e são armazenados nas variáveis `primeiroValor` e `segundoValor`.

Figura 3.2.1

```
import javax.swing.JOptionPane;

public class SomaDoisNumeros {
    public static void main (String[] args) {
        double primeiroValor, segundoValor, resultado;
        primeiroValor = Double.parseDouble(JOptionPane.showInputDialog("Digite o primeiro valor:"));
        segundoValor = Double.parseDouble(JOptionPane.showInputDialog("Digite o segundo valor:"));
    }
}
```

O método `showInputDialog` tem como parâmetro a mensagem que aparece para o usuário (sempre entre aspas). Note também que temos outro método: o `parseDouble` realizado pela classe `Double`. Isso é necessário porque o método `showInputDialog` sempre devolve uma `String`, isto é, uma sequência de caracteres, portanto temos que transformar em valor numérico, neste caso em `double`. Dá para inferir, então que precisaremos de um conversor para cada tipo:

`Boolean.parseBool` para booleanos (lógicos);
`Double.parseDouble` para pontos flutuantes de dupla precisão;
`Integer.parseInt` para inteiros.

Estamos quase lá, a entrada está resolvida, vamos para o processamento. O programa tem por objetivo somar dois números. Utilizamos para isso o operador +. O quadro 3.2.1 mostra os operadores matemáticos mais comuns e a Figura 3.2.2, o código contendo agora a etapa de processamento. Note que foram inseridos também comentários, que te ajudam a lembrar o que está acontecendo em cada trecho de código.

Quadro 3.2.1

Operador Aritmético	Operação
+	soma
-	subtração
*	multiplicação
/	divisão (depende dos operandos)
%	resto da divisão inteira

Figura 3.2.2

```
1  import javax.swing.JOptionPane; // é necessário importar a classe, poir não é parte do núcleo
2
3  public class SomaDoisNumeros {
4      public static void main (String[] args) {
5          double primeiroValor, segundoValor, resultado;
6          primeiroValor = Double.parseDouble(JOptionPane.showInputDialog("Digite o primeiro valor:"));
7          segundoValor = Double.parseDouble(JOptionPane.showInputDialog("Digite o segundo valor:"));
8          //processamento
9          resultado = primeiroValor + segundoValor;
10     }
11 }
```

A saída é simples, uma caixa de diálogo para exibir mensagens: `showMessageDialog`. Veja o código completo na Figura 3.2.3.

Figura 3.2.3

```
1  import javax.swing.JOptionPane; // é necessário importar a classe, poir não é parte do núcleo
2
3  public class SomaDoisNumeros {
4      public static void main (String[] args) {
5          double primeiroValor, segundoValor, resultado;
6          primeiroValor = Double.parseDouble(JOptionPane.showInputDialog("Digite o primeiro valor:"));
7          segundoValor = Double.parseDouble(JOptionPane.showInputDialog("Digite o segundo valor:"));
8          //processamento
9          resultado = primeiroValor + segundoValor;
10         //saída
11         JOptionPane.showMessageDialog(null, "soma = " + resultado);
12     }
13 }
```

Note que o primeiro parâmetro é a palavra reservada **`null`**, a qual discutiremos mais adiante. O segundo parâmetro é a sentença que se deseja exibir; é possível exibir uma mensagem simples, sempre entre aspas ou uma mensagem concatenada a valores de variáveis, utilizando-se o operador + (linha 11).

Seria interessante que você refizesse o programa anterior, para ver se entendeu todas as suas linhas. A seguir, tente fazer os exercícios a seguir.

Exercícios: Desenvolver um programa em Java para resolver os seguintes problemas:

1) Ler a cotação do dólar e a quantidade de dólares. Converter para real e mostrar o resultado.
2) Ler 4 números, calcular o quadrado para cada um, somar todos os quadrados e mostrar o resultado.
3) Calcular o pagamento de comissão de vendedores de peças, levando-se em consideração que sua comissão será de 5% do total da venda e que você tem os seguintes dados: preço unitário da peça e quantidade vendida.
4) Ler um valor inteiro e exibir seu antecessor.
5) Ler as dimensões de um retângulo (base e altura), calcular e escrever a área do retângulo.
6) Ler a idade de uma pessoa expressa em anos e exibir expressa em dias (considere que um ano tem 365 dias.
7) Ler a idade de uma pessoa expressa em anos, meses e dias e exibir a idade dessa pessoa expressa apenas em dias. Considerar ano com 365 dias e mês com 30 dias.
8) Ler o número total de eleitores de um município, o número de votos brancos, nulos e válidos. Calcular e escrever o percentual que cada um representa em relação ao total de eleitores.
9) Ler o salário mensal atual de um funcionário e o percentual de reajuste. Calcular e exibir o valor do novo salário.
10) O custo de um carro novo ao consumidor é a soma do custo de fábrica com a porcentagem do distribuidor e dos impostos (aplicados ao custo de fábrica). Supondo que o percentual do distribuidor seja de 28% e os impostos de 45%, ler o custo de fábrica de um carro, calcular e escrever o custo final ao consumidor.
11) Uma revendedora de carros usados paga a seus funcionários vendedores um salário fixo por mês, mais uma comissão também fixa para cada carro vendido e mais 5% do valor das vendas por ele efetuadas. Ler o número de carros por ele vendidos, o valor total de suas vendas, o salário fixo e o valor que ele recebe por carro vendido. Calcular e exibir o salário final do vendedor.