**Roteiro 2**

**Parte 1 (roteiro2.parte1) – Tipos de variáveis e entrada de dados:**

O Java é uma linguagem de programação fortemente tipada. Isso significa que em Java, os tipos das variáveis são rigidamente definidos.

No Java temos os tipos primitivos:

**int**: Armazena valores inteiros de 32 bits.

**float**: Armazena números de ponto flutuante de 32 bits.

**double**: Armazena números de ponto flutuante de 64 bits.

**char**: Armazena um caractere Unicode de 16 bits.

**boolean**: Armazena valores verdadeiro ou falso.

**long**: Armazena valores inteiros de 64 bits.

**short**: Armazena valores inteiros de 16 bits.

**byte**: Armazena valores inteiros de 8 bits.

E os tipos de Referência (Objetos):

**String**: Armazena sequências de caracteres.

**Array**: Armazena coleções de elementos do mesmo tipo.

**Classe personalizada**: Tipos definidos pelo programador, como classes que você cria.

1 – Crie o pacote **roteiro2.parte1** com a classe **Programa01** conforme o código abaixo. Observe que existem algumas variáveis declaradas com os tipos mais utilizados. Veja que uma das variáveis está sendo exibido em tela. Faça testes com as outras variáveis.

public class Programa01 {

public static void main(String[] args) {

int idade = 18;

char letra = 'a';

float numeroFloat = 3.14159f;

double numeroDouble = 2.71828;

String nome = "Issac Newton";

System.out.println("Exibir : " + numeroFloat);

}

}

2 – Crie agora uma classe no mesmo pacote chamado **Programa02**. Neste programa vamos capturar a entrada de dados do usuário, e para isso precisaremos importar a classe **Scanner** do Java. Implemente a classe Programa02 conforme o código abaixo

import java.util.Scanner;

public class Programa02 {

public static void main(String[] args) {

Scanner entrada = new Scanner(System.in);

System.out.println("Informe o seu nome");

String nome = entrada.nextLine();

System.out.println("Nome informado : " + nome);

entrada.close();

}

}

3 – Faça os devidos testes rodando o Programa02 e teste como seria uma entrada de dados de um tipo numérico. Poderíamos por exemplo solicitar a idade do usuário.

**OBS.:** a classe Scanner faz uso de conceitos de OO que serão melhor detalhados posteriormente.

**Parte 2 (roteiro2.parte2) – Entrada de dados - Gráfica:**

1 – Crie o pacote **roteiro2.parte2** com a classe **Programa03** conforme o código abaixo. Neste programa vamos capturar a entrada de dados do usuário com o uso de um pacote gráfico. Teremos que importar também o JOptionPane.

import javax.swing.JOptionPane;

public class Programa03 {

public static void main(String[] args) {

System.out.println("Informe o seu nome");

String nome = JOptionPane.showInputDialog("Informe seu nome ");

System.out.println("Nome informado : " + nome);

String strIdade = JOptionPane.showInputDialog("Informe sua idade ");

int idade = Integer.parseInt(strIdade);

System.out.println("Idade informada : " + idade);

}

}

2 – Qual a diferença percebida entre o input das variáveis nome e idade?

Na parte da idade a string “strIdade” é convertida para um número inteiro na parte

int idade = Integer.parseInt(strIdade);

Isso acontece, pois a idade é um número. O que não acontece com o nome, que de fato é uma string.

3 – Observe que no Programa03 apenas o input foi feito em ambiente gráfico, mas o output permanece no console. Substitua o comando **System.out.println( ....)** pelos respectivos comandos abaixo

JOptionPane.showMessageDialog(null, "Nome informado : " + nome);

JOptionPane.showMessageDialog(null, "Idade informada : " + idade);

**OBS.:** Neste primeiro momento não iremos nos aprofundar nos recursos gráficos para entrada e saída de dados. Em geral utilizaremos o Scanner para as nossas atividades.

**Parte 3 (s) – Estruturas de Controle**

Em Java, os comandos de controle (if, for, while, switch) são similares aos comandos em C e outras linguagens.

* **Estruturas Condicionais**
  + Exemplo da utilização do comando **if** em Java

Texto

Descrição gerada automaticamente

* + Exemplo da utilização do comando **switch** em Java

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

* **Estruturas de Repetição**
  + Exemplo da utilização do comando **for** em Java

Texto

Descrição gerada automaticamente

* + Exemplo da utilização do comando **while** em Java

Texto

Descrição gerada automaticamente com confiança média

* + Exemplo da utilização do comando **do while** em Java

Texto

Descrição gerada automaticamente

1 – Crie o pacote **roteiro2.parte3** com a classe **Programa04** conforme o código abaixo, analisando e testando o que foi implementado.

import java.util.Scanner;

public class Programa04 {

public static void main(String[] args) {

Scanner entrada = new Scanner(System.in);

int cont = 0;

while (cont < 5) {

System.out.println("Informe uma nota ");

int nota = entrada.nextInt();

if (nota >= 7) {

System.out.println("Aprovado");

}

else {

System.out.println("Reprovado");

}

cont++;

}

}

}

2 – Crie a classe **Programa05** no mesmo pacote de forma semelhante ao Programa04. Este novo programa deve apresentar o mesmo resultado anterior. Sendo que neste caso, use o **for** como estrutura de repetição.

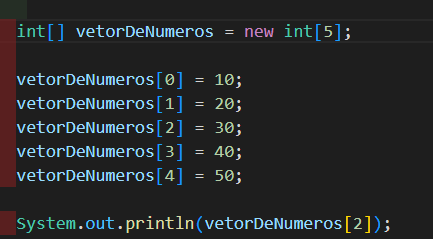
3 – Supondo agora que o input de notas não tenha uma quantidade definida e dependa de um flag. Ou seja, o usuário precisaria indicar algum valor que encerre o input de notas (exemplo flag = -1). Crie uma classe **Programa06** semelhante ao Programa05 no mesmo pacote, sendo que este novo programa deve atender os requisitos desta questão.

4 – Ainda no mesmo pacote crie a classe **Programa07** idêntico ao Programa06, e faça com que o programa apresente a média das notas que foram inseridas. Atenção para o resultado que deve ser apresentado (Exemplo : Se metade das notas inseridas tem o valor 7 e a outra metade tem o valor 8, a média esperada é 7,5).

**Parte 4 (roteiro2.parte4) – Vetores Simples (Estáticos)**

Vetores estáticos em Java funcionam da mesma forma como em outras linguagens.

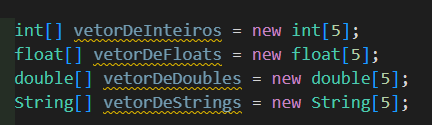
Abaixo um exemplo de declaração e utilização desse tipo de vetor em Java:



O output do exemplo acima é : 30

Na primeira linha podemos ver a forma como se declara um vetor estático de números inteiros. Em seguida a atribuição de valores, utilizados os índices do vetor e na última linha, um comando de saída, escolhendo uma das posições do vetor.

Dessa mesma forma se declara vetores estáticos de outros tipos:



1 – Crie o pacote **roteiro2.parte4** com a classe **Programa08** conforme o código abaixo e analise e teste o que foi implementado.

import java.util.Scanner;

public class Programa08 {

public static void main(String[] args) {

Scanner entrada = new Scanner(System.in);

int[] vetorNotas = new int[5];

for (int cont = 0; cont < 5; cont++){

System.out.println("Informe uma nota ");

int nota = entrada.nextInt();

vetorNotas[cont] = nota;

}

}

}

2 – Ainda na classe **Programa08,** depois do laço de repetição já existente, implemente um novo laço de repetição apenas para exibir os valores armazenados no vetor. Verifique se as notas exibidas estão corretas.

3 – Crie uma nova classe **Programa09** idêntica ou Programa08, e implemente uma solução que armazene no vetor apenas notas consideradas válidas. Ou seja, notas inseridas com o valor de 0 a 10. Caso não seja uma nota válida, o programa deve informar que a nota é inválida, e não deve armazenar esta nota. Atenção : O programa também deve garantir que todo o vetor será preenchido com notas válidas.

4 – Utilizando a classe Programa07 implementado no pacote roteiro2.parte3, seria possível implementar uma solução que apresentasse não somente a média, mas quantas notas ficaram acima da média ? Sim ? Não ? Como ? Por quê ?

Sim, seria possível implementar uma solução que, além de calcular a média, também contabilizasse quantas notas ficaram acima da média. Para isso, precisaria armazenar todas as notas inseridas pelo usuário e, após calcular a média, percorrer as notas para contar quantas estão acima dessa média.

**Parte 4 (roteiro2.parte4) – Exercício**

Objetivo: Criar um programa de adivinhação em que o usuário tenta adivinhar um número secreto.

1. Gere um número secreto aleatório entre 1 e 100. Utilize para isso a classe Random do Java (java.util.Random):

Random random = new Random();

Int numeroSecreto = random.nextInt(100)+1;

1. Peça ao usuário para adivinhar o número.
2. Use um loop **while** para repetir até o usuário acertar.
3. Dentro do loop, use o comando **if** para verificar se o palpite é correto, alto ou baixo.
4. Forneça feedback ao usuário sobre o palpite, e indique se o palpite é maior ou menor que o número secreto.
5. Quando o usuário acertar, termine o loop.