LTP 1 – Java Básico

Aula 4

Tópicos

1 Solução dos exercícios da aula anterior

a) Crie uma classe com um método que lê dois valores inteiros e distintos e informe qual é o maior. Esse método deve ser chamado pelo método main de outra classe do mesmo projeto.

```
import java.util.Scanner;
public class Calculo {
    public void leValores(){
        int valor1,valor2;
        Scanner aux = new Scanner(System.in);
        System.out.println("Entre com dois valores distintos");
        valor1 = aux.nextInt();
        valor2 = aux.nextInt();
        if(valor1>valor2)
            System.out.println(valor1+" maior que "+valor2);
        else
            System.out.println(valor2+" maior que "+valor1);
    }
}
```

```
public class Exemplo {
    public static void main(String[] args)
    {
        Calculo ref = new Calculo();
        ref.leValores();
    }
}
```

Na tela:

Entre com dois valores distintos
2
3
3 maior que 2

b) Crie uma classe com um método que lê um número e determina se o mesmo é par ou ímpar. Um número é par se for divisível exato por 2, ou seja, se o referido número for dividido por 2 e for par o resto será zero. Esse método deve ser chamado pelo método main de outra classe do mesmo projeto.

```
import java.util.Scanner;
public class Calculo {
    public void parImpar()
    {
        int num;
        Scanner var = new Scanner(System.in);
        System.out.println("Entre com um valor");
        num = var.nextInt();
        if(num%2 ==0)
            System.out.println(num+" é par");
        else
            System.out.println(num+" é ímpar");
    }
}
```

```
public class Exemplo
{
    public static void main(String[] args)
    {
        Calculo var = new Calculo();
        var.parImpar();
    }
}
```

Na tela:

Entre com um valor
7
7 é ímpar

c) Crie uma classe com um método que lê o nome e quatro notas de um aluno durante o semestre, calcula a média aritmética dessas notas. Se a nota for maior ou igual a (seis) o aluno será aprovado. Se a média for menor que seis, o programa deverá ler a nota do exame final e calcular a nova média, da seguinte forma: Média Final = (Média+Exame Final)/2. Nesse caso, para ser aprovado a média final deverá ser maior ou igual a cinco. Esse método deve ser chamado pelo método main de outra classe do mesmo projeto.

```
import java.util.Scanner;
public class MediaAritmetica {
    public void calcula()
    {
        float n1,n2,n3,n4,media,exameFinal,mediaFinal;
        Scanner valor = new Scanner(System.in);
        System.out.println("Digite 4 valores");
        n1=valor.nextFloat();
        n2=valor.nextFloat();
        n3=valor.nextFloat();
        n4=valor.nextFloat();
        media=(n1+n2+n3+n4)/4;
        if(media>=6)
            System.out.printf("Media = %.1f\n Aprovado\n",media);
        else
        {
            System.out.println("Qual a nota do Exame Final?");
            exameFinal=valor.nextFloat();
            mediaFinal=(media+exameFinal)/2;
```

```
            if(mediaFinal>=5)
                System.out.printf("Media = %.1f\n Aprovado\n",media);
            else
                System.out.printf("Media = %.1f\n REPROVADO\n",media);
        }
    }
}
```

```
public class Exemplo
{
    public static void main(String[] args)
    {
        MediaAritmetica var = new MediaAritmetica();
        var.calcula();
    }
}
```

Na tela:

Digite 4 valores
5
6
7
2
Qual a nota do Exame Final?
7
Media = 5,0
Aprovado

d) Crie uma classe com um método que lê o comprimento de 3 pedaços de madeira e verifica se os mesmos podem formar um triângulo, ou seja, se cada lado é menor que a soma dos outro dois. Esse método deve ser chamado pelo método main de outra classe do mesmo projeto.

A < B + C
B < A + C
C < A + B

```
import java.util.Scanner;
public class Triangulos {
    public void testaLados()
    {
        Scanner var = new Scanner(System.in);
        int a,b,c;
        System.out.println("Entre com o valor dos lados do triângulo");
        a = var.nextInt();
        b = var.nextInt();
        c = var.nextInt();
        if(a<b+c && b<a+c && c<a+b)
            System.out.println("Pode formar um triângulo");
        else
            System.out.println("Não pode formar um triângulo");
    }
}
```

```
public class Exemplo
{
    public static void main(String[] args)
    {
        Triangulos var = new Triangulos();
        var.testaLados();
    }
}
```

Na tela:

Entre com o valor dos lados do triângulo
2
2
7
Não pode formar um triângulo

e) Crie uma classe com um método que lê o comprimento de 3 pedaços de madeira e verifica se os mesmos podem formar um triângulo. Esse método deve ser chamado pelo método main de outra classe do mesmo projeto. Se formar um triângulo, determinar o tipo de triângulo:

Equilátero = 3 lados iguais
Isósceles = 2 lados iguais
Escaleno = 3 lados diferentes

```
import java.util.Scanner;
public class Triangulos {
    public void testaLados()
    {
        Scanner var = new Scanner(System.in);
        int a,b,c;
        System.out.println("Entre com o valor dos lados do triângulo");
        a = var.nextInt();
        b = var.nextInt();
        c = var.nextInt();
        if(a<b+c && b<a+c && c<a+b)
        {
            if(a==b&&b==c&&a==c)
                System.out.println("Pode formar um triângulo equilátero");
            else
            {
                if(a==b||a==c||b==c)
                    System.out.println("Pode formar um triângulo isósceles");
                else
                    System.out.println("Pode formar um triângulo escaleno");
            }

        }
        else
            System.out.println("Não pode formar um triângulo");
    }
}
```

```
public class Exemplo
{
    public static void main(String[] args)
    {
        Triangulos var = new Triangulos();
        var.testaLados();
    }
}
```

Na tela:

Entre com o valor dos lados do triângulo
3
3
5
Pode formar um triângulo isósceles

2 Um método com passagem de parâmetro

Num carro podemos mandar uma mensagem para método acelerador fazer o carro andar mais rápido. Mas quanto o carro deve acelerar? Essas informações adicionais são conhecidas como parâmetros. Você já trabalhou com métodos que exigem passagem de parâmetros, sem perceber: System.out.println( ), que pede que você informe quais dados serão exibidos na tela.

2.1 Formato Geral

```
public void nomeDoMétodo(tipo parâmetro)
{
      instruções
}
```

Onde:

- Tipo pode ser um dos tipos primitivos (int, float, char, double,...) ou o nome de uma classe.
- Parâmetro é o nome da variável que irá receber uma cópia do valor.

Para chamar esse método deverá usar algo assim:

nomeDoMétodo(**parâmetro)**

Onde:

- O parâmetro deverá ser do mesmo tipo e com a mesma quantidade de parâmetros definidos na declaração do referido método.

2.2 Exemplo

```
public class BoasVindas//salva no arquivo BoasVindas.java
{
    public void mostraMensagem(String curso)
    {
       System.out.println("Bem vindo ao curso de Java do "+curso);
    }
}
public class Exemplo //salva no arquivo Exemplo.java
{
    public static void main(String args[ ])
    {
      BoasVindas msg=new BoasVindas( );
       msg.mostraMensagem ("Salvador");//chamada do método mostraMensagem
    }
}
```

Observações:

- Os tipos de informações a serem passadas na chamada do método, deve ser coerente com os tipos especificados na declaração do referido método;
- É um erro não passar um parâmetro para o método, quando o mesmo é exigido;
- A cada parâmetro você especificar o tipo e um identificador (nome da variável);
- Um método pode especificar mais de um parâmetro, separando cada um deles do próximo com uma vírgula.

3 Variáveis Locais e Variáveis de instância

Variáveis locais: são aquelas declaradas no corpo de um método e só podem ser utilizadas nesse método. Ao fim da execução do método os valores das duas variáveis locais são perdidos.

As variáveis de instância são aquelas que são declaradas dentro de uma classe, mas fora do corpo das declarações dos métodos dessa mesma classe, também conhecidas como campos.

A maioria das declarações de variável de instância é precedida pela palavra-chave private, que também é um modificador de acesso, assim como a public. Nesse caso as variáveis (ou métodos) declarados como private só são acessíveis a métodos da classe onde foram declarados.

3.1 Exemplo (com erro para análise)

```
public class Calculo
{
   private double resultado;//resultado é uma variável de instância
   public void somaValores(double x,double y)//x e y são variáveis locais
   {
       resultado = x+y;
       System.out.println(x+" + "+y+" = "+resultado);
   }
}
```

```
public class Exemplo
{
   public static void main(String args[])
   {
       Calculo soma;
       soma = new Calculo( );
       soma.resultado = 5;
       soma.somaValores(2.0,3.0);
   }
}
```

Na Tela:
resultado has private access in Calculo
Observação:

- Podemos colocar as variáveis de instância (os campos de uma classe) em qualquer lugar dentro da classe e fora dos métodos, mas recomenda-se fazer isso logo no começo da classe para facilitar a leitura.

4 Métodos com retorno

Nem todo método tem retorno do tipo void. Quando um método que especifica um tipo de retorno for chamado e completar sua tarefa, o método retornará uma informação para o seu método chamador. O tipo de retorno é especificado por uma palavra chave ou nome de classe na declaração do método, estabelecendo o tipo de valor que ele pode devolver.

4.1 Formato Geral

public **tipo** nomeDoMétodo( )
{
instruções
return **dado**;
}

Onde:

- Tipo é o tipo de retorno que o referido método irá retornar ao finalizar sua execução;
- Dado é aquilo que o método retorna para quem o chamou.

4.2 Exemplo

```
public class Calculo {
    public float divisaoNormal(float a, float b){
        float valor;
        valor = a/b;
        return valor;
    }
```

```
    public float restoDivisao(float a, float b)
    {
        float valor;
        valor = a%b;
        return valor;
    }
}
```

```
import java.util.Scanner;
public class Exemplo
{
    public static void main(String[] args)
    {
        Calculo var = new Calculo();
        Scanner x = new Scanner(System.in);
        float valor1,valor2,opcao;
        System.out.println("Entre com dois valores");
        valor1 = x.nextFloat();
        valor2 = x.nextFloat();
        System.out.println("Obter a divisão normal(1) "
                + "ou o resto da divisão(2)?");
        opcao = x.nextFloat();
        if(opcao==1)
            System.out.println("Resultado = "
                      +var.divisaoNormal(valor1,valor2));
        else
            System.out.println("Resultado = "
                      +var.restoDivisao(valor1,valor2));
    }
}
```

Na tela:
Entre com dois valores
3
2
Obter a divisão normal(1) ou o resto da divisão(2)?
1
Resultado = 1.5

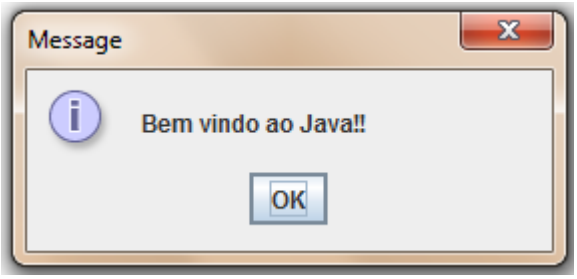
5 Introdução a GUI – usando caixas de diálogo

Embora os programas apresentados até aqui exibam na própria janela da IDE (por exemplo o NetBeans), muito aplicativos Java utilizam caixas de diálogo para exibir a saída. Para isso podemos utilizar a classe JOptionPane do pacote javax.swing.

5.1 Exemplo 1

```
import javax.swing.JOptionPane;
public class Exemplo
{
    public static void main(String args[ ])
    {
        JOptionPane.showMessageDialog(null,"Bem vindo ao Java!!\n");
    }
}
```

Na tela:



O pacote javax.swing possui muitas classes que ajudam a criar interfaces gráficas com o usuário (Graphical User Interfaces – GUI) para aplicativos. O método showMessagePane é **static** da classe JOptionPane, por isso não exigem explicitamente a criação de um objeto. Então no caso de métodos static basta fazer o seguinte para usá-los:

NomeDaClasse.NomeDoMétodo(argumentos);

5.2 Outros métodos da classe JOptionPane

- Para mostrar um string no centro da tela
  `JOptionPane.showMessageDialog(null,String);`

- Para ler um string e retorna um string
  `JOptionPane.showInputDialog(String);`

- Para fazer uma pergunta e obter uma das seguintes respostas: retorna 0 se a resposta à pergunta do string for sim, //retorna 1 se a resposta for não e retorna 2 se for cancel
  `JOptionPane.showConfirmDialog(null, String);`

5.3 Convertendo de String para Número

Mesmo com JOptionPane.showInputDialog você lê um string, porém para utilizar em suas contas e cálculos, deverá converter esse string para double ou int, da seguinte forma:

- int x = Integer.parseInt(String);
- float x = Float.parseFloat(String);
- double x = Double.parseDouble(String);

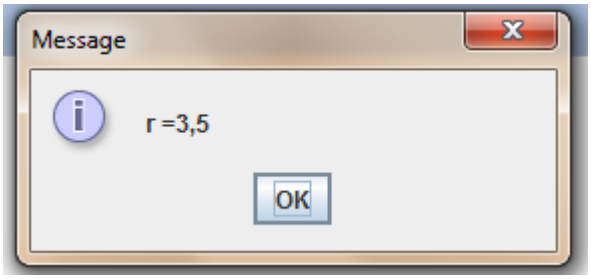
5.4 Convertendo de número e texto para String

Você também pode usar o método format da classe String. Esse método serve para "formatar" a String s usando os objetos passados como argumentos. Basicamente ele usa marcadores (%) dentro da String s que serão substituidos pela representação dos objetos fornecidos. Esse marcadores começam com um % e são seguidos por códigos que indicam como os objetos devem ser representados. "%s", por exemplo, significa para usar a represantação como um String (toString), "%d" para inteiros, "%f" para decimais.

5.4.1 Exemplo 1

```
import javax.swing.JOptionPane;
public class Ex1
{
    public static void main(String args[])
    {
        String texto;
        float r=3.5f;
        texto = String.format("r =%.1f", r);
        JOptionPane.showMessageDialog(null,texto);
    }
}
```

Na tela:

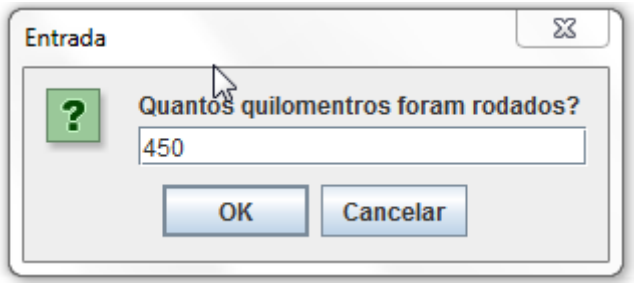


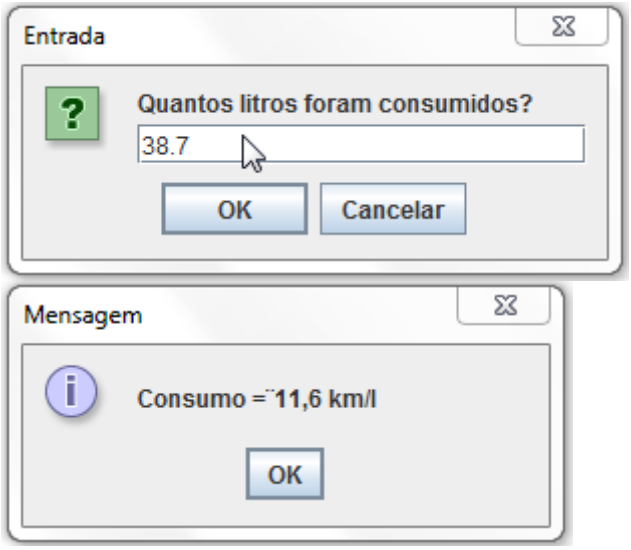
5.4.2 Exemplo 2

```
import javax.swing.JOptionPane;
public class MinhasJanelas {
    public void exemploGUI()
    {
       String aux1,aux2,saida;
       double consumo;
       aux1 = JOptionPane.showInputDialog("Quanto quilometros foram rodados?");
       aux2 = JOptionPane.showInputDialog("Quanto litros foram consumidos?");
       consumo = Double.parseDouble(aux1)/Double.parseDouble(aux2);
       saida = String.format("Consumo = %.1f km/l",consumo);
       JOptionPane.showMessageDialog(null,saida);
    }
}
```

```
public class Exemplo
{
    public static void main(String[] args)
    {
        MinhasJanelas var = new MinhasJanelas();
        var.exemploGUI();
    }
}
```

Na tela:





6 Novos Exercícios

a) Crie uma classe com estes métodos:

- um método que recebe o comprimento de três pedaços de madeira, verifica se os mesmos podem formar um triângulo e retorna 0 em caso afirmativo e 1 em caso negativo;
- um outro método que verifica e informa na tela qual o tipo de triângulo que pode ser formado.

Crie uma segunda classe, com o método main, que fará a leitura do comprimento de três pedaços de madeira e chama o método que verifica se os mesmos podem formar um triângulo ou não. Se puder formar um triângulo, esse método deverá chamar o outro que verifica que tipo de triângulo pode ser formado, caso contrário deverá informar a seguinte frase "não pode formar um triângulo". Use os métodos printf ou println da classe System e os métodos da classe Scanner.

b) Refaça o exercício acima usando os métodos da classe JOptionPane.

c) Crie uma classe chamada Fatura para que uma loja de suprimentos em informática possa apresentar a fatura de um produto vendido na loja. Essa classe deverá ter uma variável de instância denominada totalFatura. Na referida classe deverá ter um método que recebe estas informações:

- Número da fatura (String)
- Nome do produto (String)
- Quantidade de produtos (int)
- Preço de cada produto (double)

Então, calcule o total da fatura (preço x quantidade) e guarde na variável de instância criada anteriormente. Nessa mesma classe, crie um segundo método que mostra uma tela com todos os itens da fatura, inclusive o total da mesma.

.

d) Crie uma classe com um método chamdo maiorValor, que recebe 2 valores e retorna o maior. Cria uma classe com o método main que lê dois números pelo teclado, chama o método maiorValor e mostra uma frase informando que é o maior valor, a partir do retorno do método maiorValor. Use os métodos da classe JOptionPane.