

## Capítulo - 06

### 6.1- Uso da classe Scanner

Vimos no capítulo 4 a entrada de dados direto no prompt de comandos, mas como você deve saber isto não é comum. Normalmente ninguém que executa um programa insere valores no prompt.

Outra possibilidade de entrada de informações via teclado refere-se ao uso da classe Scanner, disponível a partir da versão J2SE 5.0. A grande vantagem desta classe é a possibilidade de promover a entrada já convertendo os valores para o formato desejado. Com esta classe poderemos ler a linha inteira digitada pelo usuário e utilizá-la, por exemplo, em uma variável para posterior processamento.

Veja o código abaixo:

---

```
1 import java.util.Scanner;
2 public class EntradaComScanner {
3
4     public static void main(String[] args) {
5         float nota1 = 0, nota2 = 0, trabalho = 0, media = 0;
6
7         System.out.println("Entre com a nota 1");
8         Scanner dado = new Scanner(System.in);
9         nota1 = dado.nextFloat();
10
11        System.out.println("Entre com a nota 2");
12        Scanner dado1 = new Scanner(System.in);
13        nota2 = dado1.nextFloat();
14
15        System.out.println("Entre com a nota do Trabalho");
16        Scanner dado2 = new Scanner(System.in);
17        trabalho = dado2.nextFloat();
18
19        media = ((nota1 + nota2 + trabalho) / 3);
20
21        System.out.println("Média " + media);
22    }
23 }
```

---

Deste ponto em diante utilizaremos numeração de linhas e será altamente recomendável o uso da ferramenta NotePad++ por facilitar a inteligibilidade do código e permitir a numeração da linhas que serão comentadas.

**Na linha 1** invocamos através da cláusula `import` a classe `Scanner` que está contida no pacote `java.util`. A diretiva `import` explica para o compilador que deve ser incluído as funcionalidades contida na classe `Scanner`.

**Na linha 5** declaramos as variáveis do tipo `float` que serão utilizadas pela classe. O tipo `float` foi escolhido por ser o tipo mais apropriado.

Responda abaixo por quê.

---

---

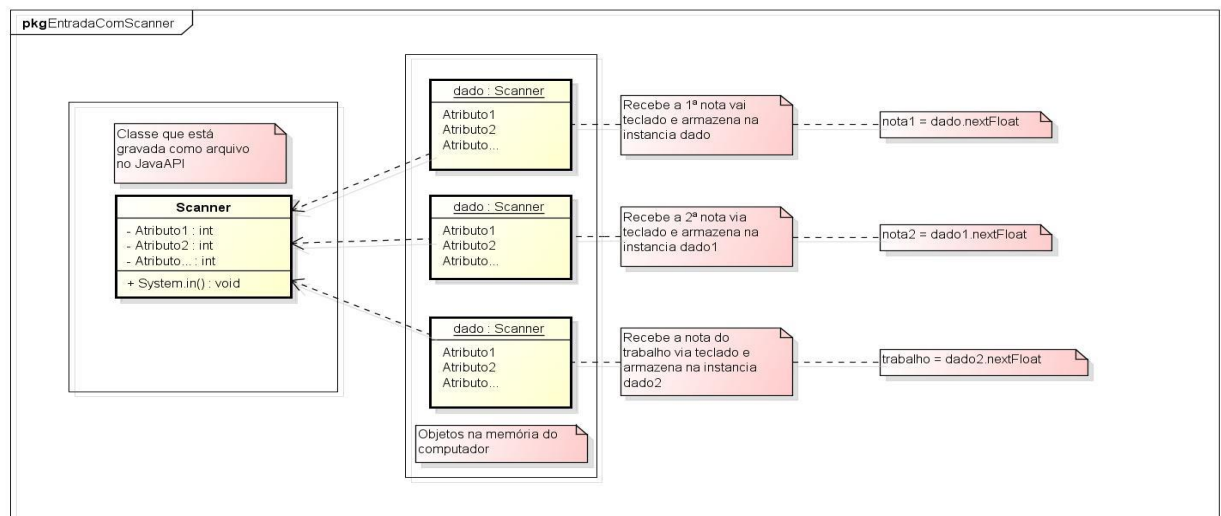
**Na linha 8** chamamos a classe `Scanner` e instanciamos ela com o nome `dado`. A classe `Scanner` requer o argumento `System.in`, que dá uma “paradinha” para o usuário digitar.

**Na linha 9** a variável `nota1` receberá o valor digitado pelo teclado através da instancia `dado` e já converterá para o tipo `float`, sim, pois como sabemos tudo que o usuário digitar pelo teclado o Java recebe como tipo `String` ou texto.<sup>7</sup>

**Na linha 12** criamos mais uma instancia da classe `Scanner` mas agora com o nome `dado1`.

**Na linha 16** criamos mais uma instancia da classe `Scanner` mas agora com o nome `dado2`.

Farei aqui uma descrição gráfica das instancias da classe **Scanner** pois entendendo este conceito será fácil entender outros assuntos relacionados a OO (orientação a objetos).

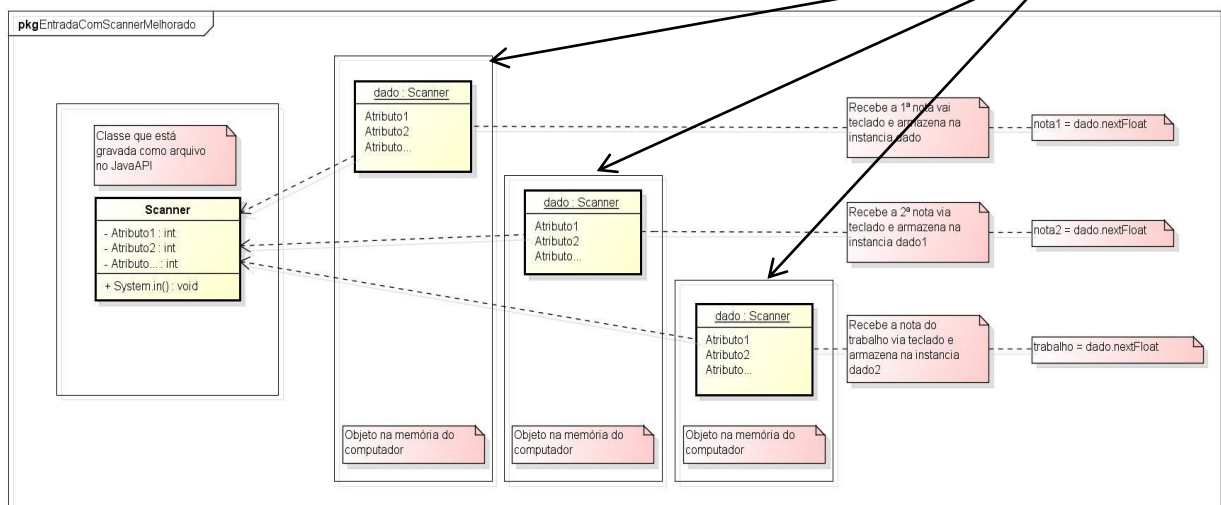


Quadro 03 – Diagrama UML do programa notas de alunos

## 6.2 - Outra forma de uso da classe Scanner

```
1 import java.util.Scanner;
2
3 public class EntradaComScannerMelhorado {
4
5     public static void main(String[] args) {
6         float nota1 = 0, nota2 = 0, trabalho = 0, media = 0;
7
8         System.out.println("Entre com a nota 1");
9         Scanner dado = new Scanner(System.in);
10        nota1 = dado.nextFloat();
11
12        System.out.println("Entre com a nota 2");
13        dado = new Scanner(System.in);
14        nota2 = dado.nextFloat();
15
16        System.out.println("Entre com a nota do Trabalho");
17        dado = new Scanner(System.in);
18        trabalho = dado.nextFloat();
19
20        media = ((nota1 + nota2 + trabalho) /3);
21
22        System.out.println("Média " + media);
23    }
24 }
```

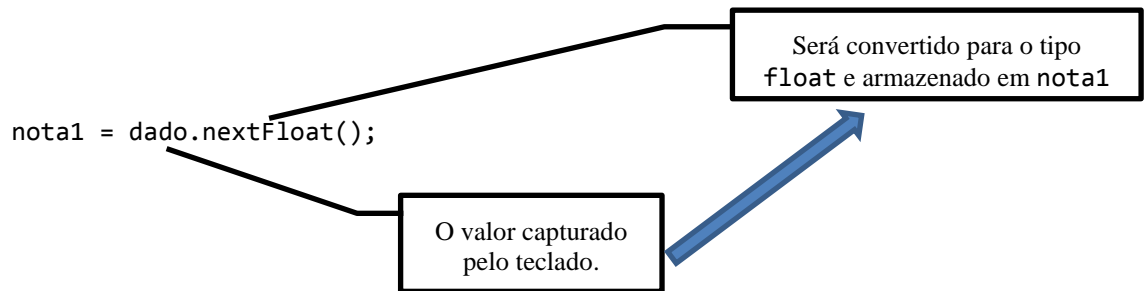
Neste caso temos apenas uma instancia da classe **Scanner**. Esta instancia vai sendo “destruída” em cada uma das perguntas para o usuário digitar notas. Ou seja, temos o seguinte diagrama.



Quadro 04 – Diagrama UML do programa notas de alunos Modificado

Perceba na classe `Scanner` que o método `nextFloat()` é utilizado para fazer a conversão para o tipo `float` que foi declarado nas variáveis `nota1`, `nota2` e `trabalho`.

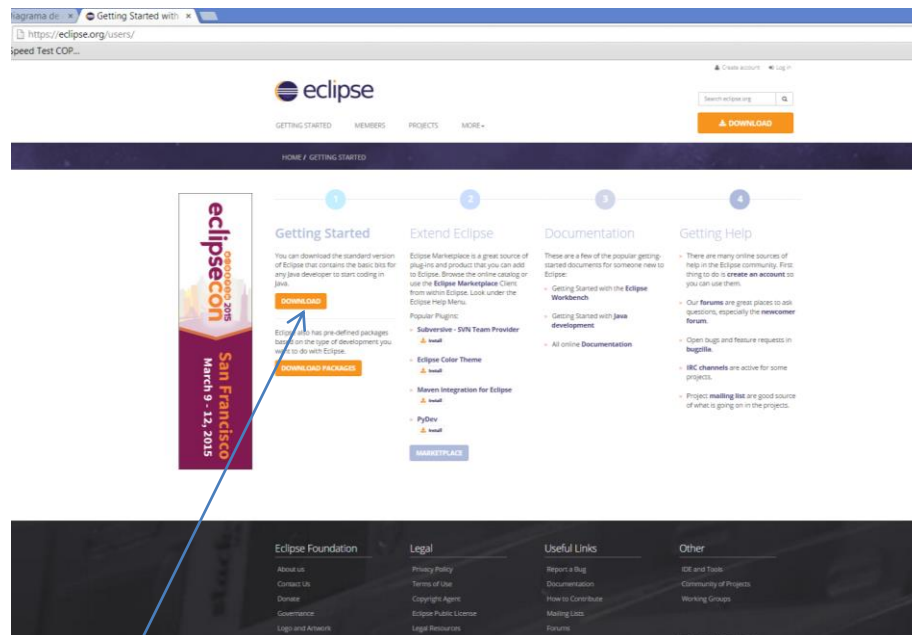
Veja novamente!



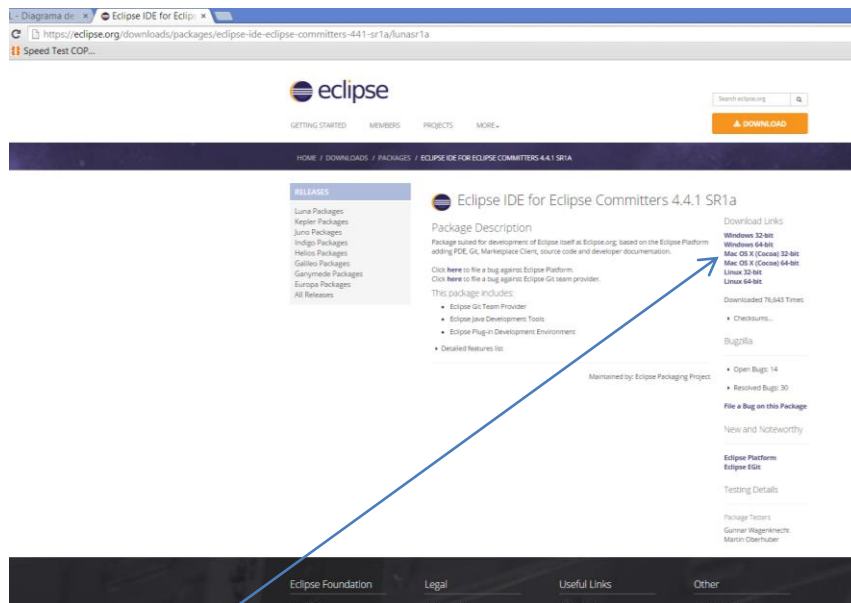
### 6.3 - Instalação do Eclipse

Deste ponto em diante sugiro o uso da IDE Eclipse que facilita em muito o desenvolvimento de aplicações em Java. Se ainda não tem esta poderosa ferramenta instalada. Siga os passos.

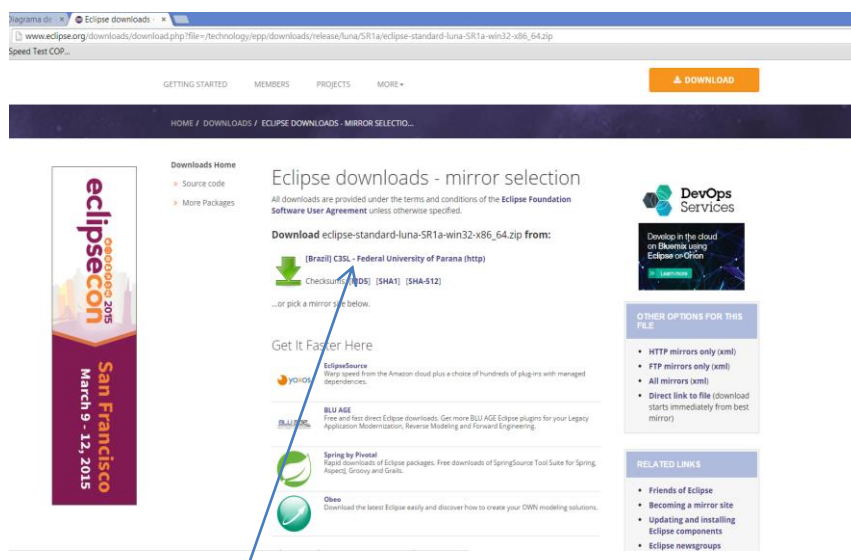
- 1) O eclipse pode ser facilmente encontrado na internet em <https://eclipse.org/> no momento da elaboração desta apostila a versão mais recente é o LUNA.



- 2) Clique em Download.



3) Seleccione o seu sistema operacional



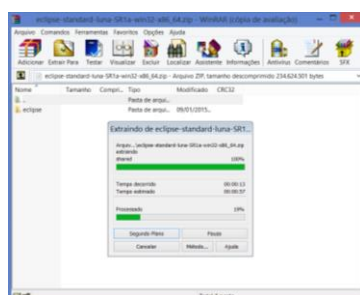
4) Clique em um dos links disponíveis.

No link seguinte talvez os desenvolvedores peçam uma doação que não é obrigatória.

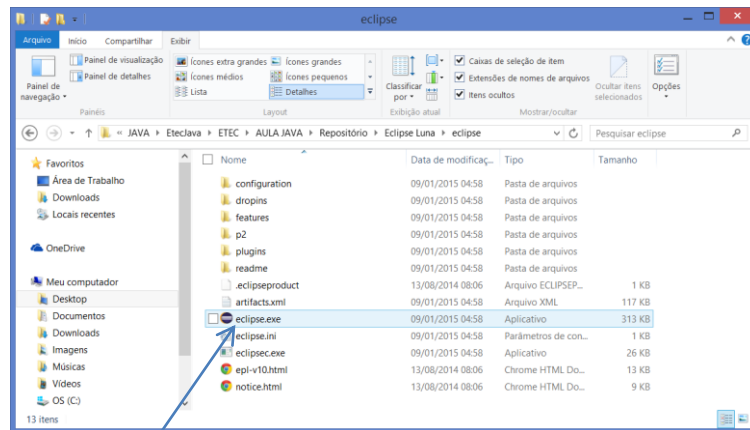
Enquanto você vai baixando o eclipse saiba que o Eclipse foi originalmente desenvolvido pela IBM em novembro de 2001 e suportado por um consorcio de fornecedores de software. A fundação Eclipse foi criada em janeiro de 2004 como uma organização sem fins lucrativos e independente chamada de comunidade Eclipse

eclipse-standard-luna-SR1a-win32-x86_64.zip	21/02/2015 23:12	Arquivo ZIP	209.679 KB
---	------------------	-------------	------------

5) Descompacte o arquivo em uma pasta.



- 6) Pasta aberta contendo os arquivos do Eclipse descompactado.

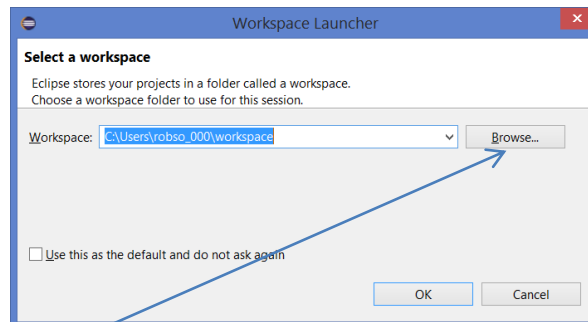


- 7) Clique duas vezes em

- 8) Dependendo da velocidade de seu computador logo você verá o SplashScreen da IDE-Eclipse

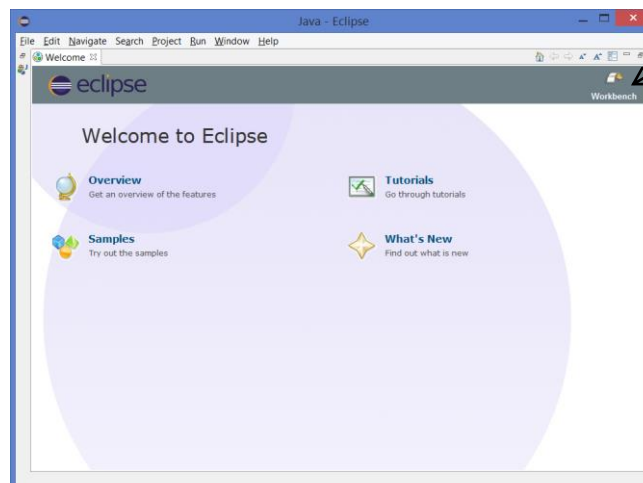


- 9) A primeira configuração a ser feita é escolher o Diretório de Trabalho para iniciar a criação das classes de seu projeto. Este diretório é chamado de Workspace.



- 10) Clique no botão Browse e selecione um local desejado, pen-drive ou uma pasta no próprio computador.

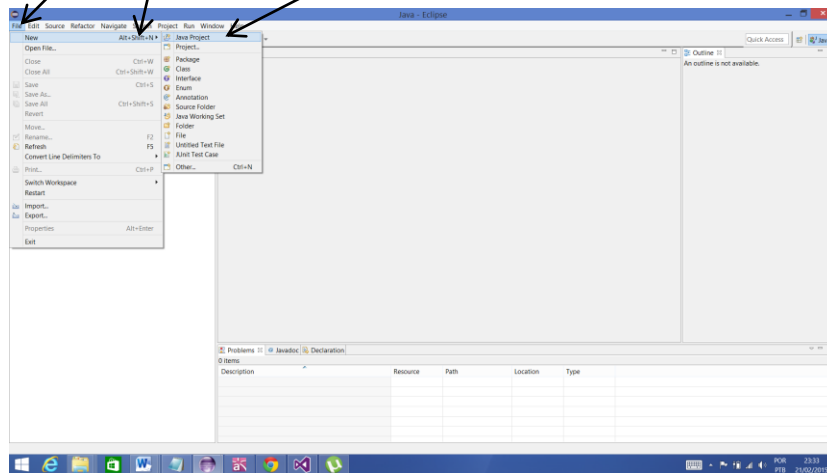
- 11) Depois de alguns instantes você estará no ambiente de desenvolvimento. Clique em Workbench



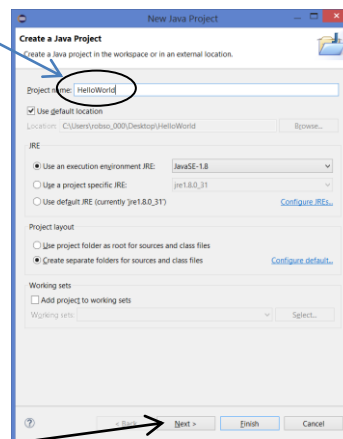
## 6.5 – Utilizando o Eclipse pela primeira vez

Agora que já baixou o programa mãos à obra!

- 1) Clique no menu File, New e selecione Java Project ou simplesmente ALT+SHIFT+N

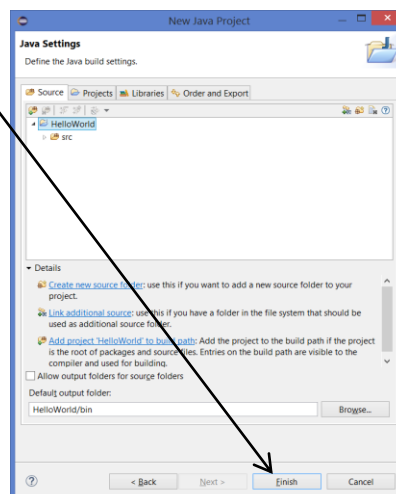


- 2) Defina um nome para seu projeto. É no projeto que ficarão armazenadas as classes. Antes do uso do eclipse nossas classes.java e os arquivos de bytecodes.class ficavam a deriva em uma pasta escolhida por nós. Agora eles ficarão sob a guarda da pasta de projeto que mantém algumas configurações de ambiente.

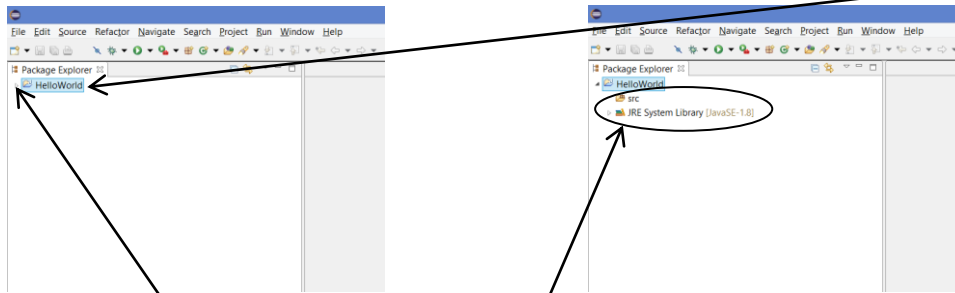


- 3) Clique em Next (próximo)


- 4) Em seguida clique em Finish

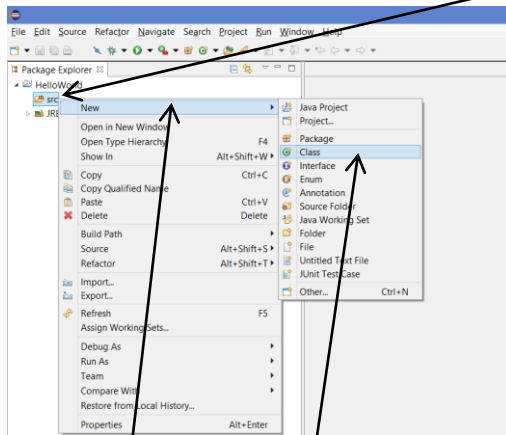


- 5) Perceba que agora aparecerá no lado da esquerda uma pasta de projeto chamada HelloWorld



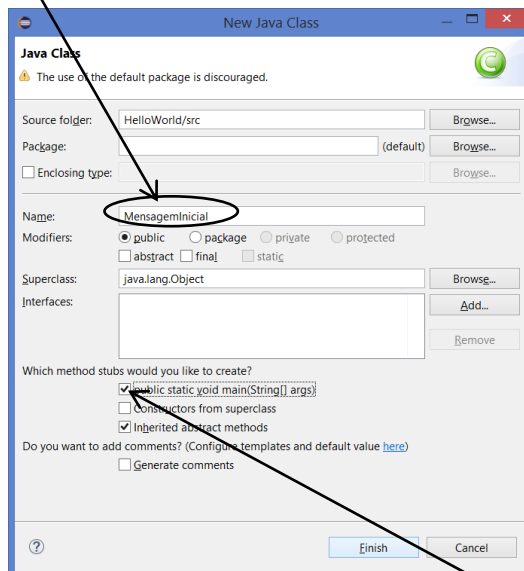
- 6) Clique na “setinha” da pasta de projeto para expandir o conteúdo.

- 7) Clique com o botão direito do mouse na pasta 



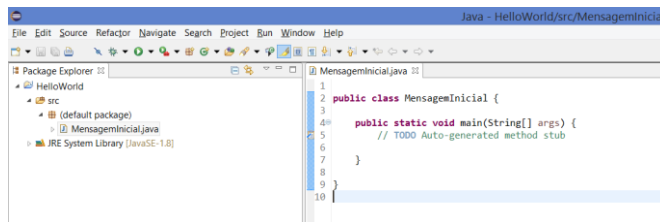
- 8) Escolha New, em seguida Class.

- 9) Dê um nome para a sua classe. Nos exemplos anteriores quando criamos as nossas primeiras classes, nomeávamos depois! Agora você deverá dar o nome para a classe antes de construí-la. Em nosso caso a classe vai se chamar MensagemInicial.



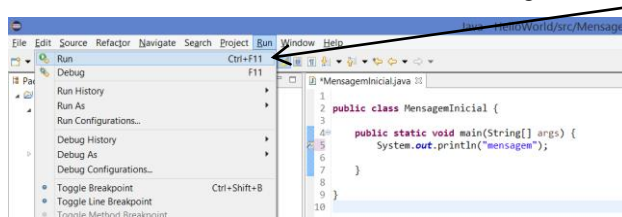
- 10) Esta será a classe principal do sistema, selecione também `public static void main(String[] args)`. Agora você não precisará mais digitar este método na classe principal o Eclipse o fará para você!

11) Clique no botão Finish.

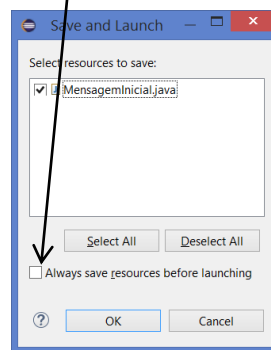


Pronto! Tudo certo para você desenvolver seus programas em Java de uma forma profissional, produtiva e mais interativa.

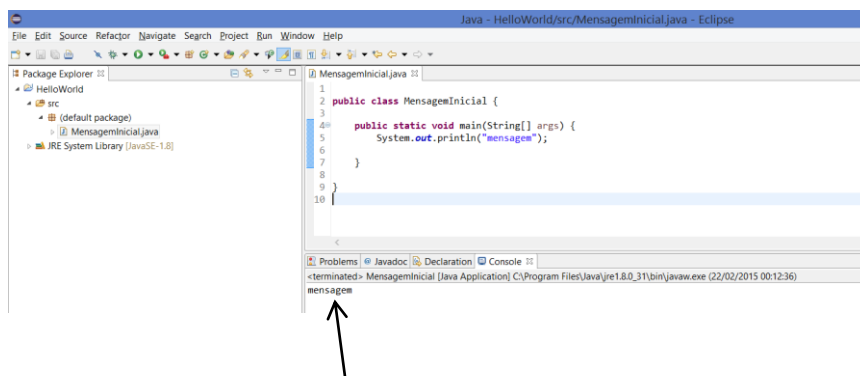
12) Para executar sua classe nada de ir para o prompt de comando compilar com Javac e rodar com o comando Java. Basta Clicar no menu Run em seguida Run novamente. Ou simplesmente CTRL+F11.



Na primeira execução o Eclipse lhe perguntará se você deseja que ele salve o projeto antes de executar Para que ele não pergunte mais basta clicar em:



Em seguida clicar no botão OK



Eis então a nossa classe em execução.

Claro que este exemplo é bem simples para você aluno que já desenvolveu várias classes mais complexas.

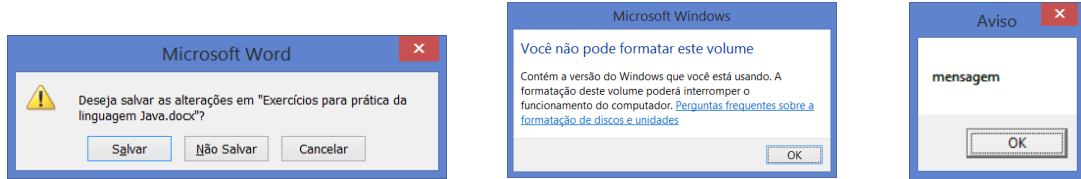
### Dica!

No Java não há problema em continuar o código na linha de baixo. Simplesmente aperte o enter e tudo bem. É claro que você só não pode quebrar palavras reservadas e nomes de métodos, instancias e variáveis.



## 6.4 - Caixa de diálogo para a Entrada e Saída de dados

A linguagem Java dispõe de uma forma gráfica para receber dados do usuário. Trata-se da utilização de caixas de diálogo. Aquelas que aparecem no Windows sempre que uma mensagem deve ser exibida. Veja.



No Java estas caixas são geradas a partir da classe `JOptionPane`. Grave bem este nome, você o usará muito nos nossos próximos exercícios. O fato de receber dados e exibir mensagens de forma gráfica dará mais motivação para você aluno programador Java. O exemplo abaixo é a classe das médias, mas desta vez, refeito com o emprego da classe `JOptionPane` para entrada e saída de dados.

```
1 import javax.swing.JOptionPane;
2
3 public class UsoJOptionPane {
4     public static void main(String[] args)
5     {
6         float nota1 = 0, nota2 = 0, trabalho = 0, media = 0;
7         String aux = "";
8
9         aux = JOptionPane.showInputDialog(null, "Entre com a nota 1");
10        nota1 = Float.parseFloat(aux);
11
12        aux = JOptionPane.showInputDialog(null, "Entre com a nota 2");
13        nota2 = Float.parseFloat(aux);
14
15        aux = JOptionPane.showInputDialog("Entre com a nota do Trabalho");
16        trabalho = Float.parseFloat(aux);
17
18        media = ((nota1 + nota2 + trabalho) / 3);
19
20        JOptionPane.showMessageDialog(null, "Média " + media);
21    }
22 }
23 }
```


**Na linha 7** declaramos uma variável chamada `aux`, apenas para receber os valores digitados pelo usuário. Ela é do tipo `String` pois como sabemos é o tipo de valor recebido pelo Java mesmo sendo de forma gráfica através do `JOptionPane`.

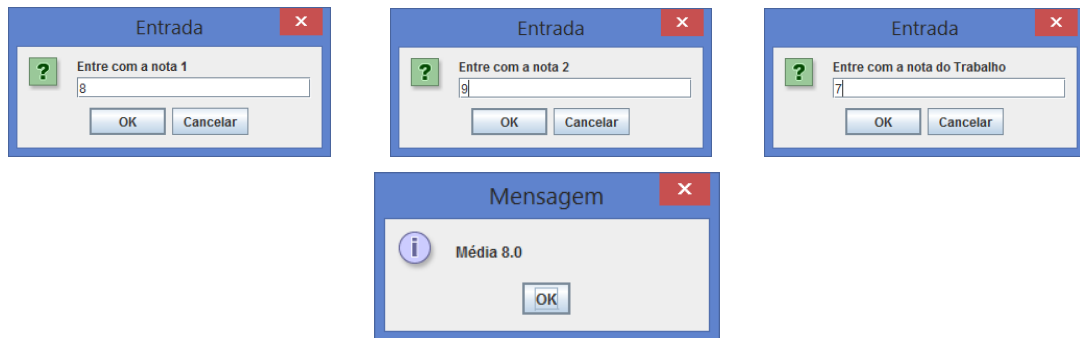
**Na linha 9** o comando `JOptionPane.showInputDialog` traz o valor digitado na caixa de diálogo que aparece na tela ao qual é atribuído com o sinal de "=" para a variável `aux`.

**Na linha 10** A variável `nota1` recebe o valor contido em `aux`. Porém antes ele é convertido para o tipo `float` através do comando `Float.parseFloat()`. Veja a tabela 03 para outras conversões.

**Na linha 11** apenas fazemos o inverso que é exibir o valor da variável `média`. Semelhante ao que fazíamos com `System.out.print`.

O valor da variável `aux` é sempre substituído a cada nova pergunta, porém antes este valor é capturado pelas variáveis `nota1`, `nota2` e `trabalho`!

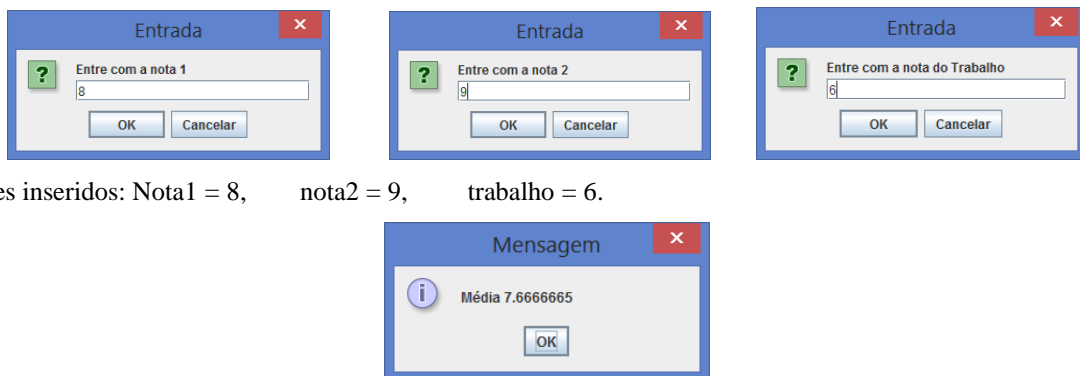
Clique em  ou CTRL+F11 no Eclipse para executar esta classe. E entre com os valores abaixo:



The first set of dialog boxes shows the following inputs:

- Entrada: Entre com a nota 1. Input: 8.
- Entrada: Entre com a nota 2. Input: 9.
- Entrada: Entre com a nota do Trabalho. Input: 7.
- Mensagem: Média 8.0.

Agora entre com estes valores:



The second set of dialog boxes shows the following inputs:

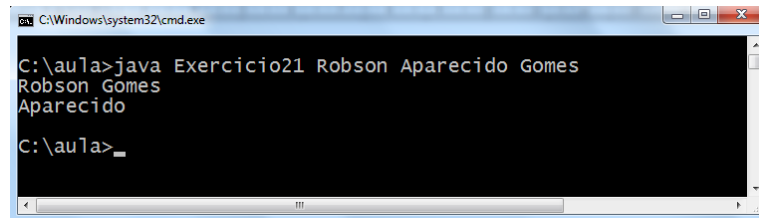
- Entrada: Entre com a nota 1. Input: 8.
- Entrada: Entre com a nota 2. Input: 9.
- Entrada: Entre com a nota do Trabalho. Input: 6.
- Mensagem: Média 7.6666665.

Valores inseridos: Nota1 = 8, nota2 = 9, trabalho = 6.

**Obs:** Dependendo das notas inseridas, a média aparecerá com muitos dígitos após a vírgula, como no exemplo abaixo. Não se preocupe, pois aprenderemos a tratar estes valores de forma que apareça 7,7.

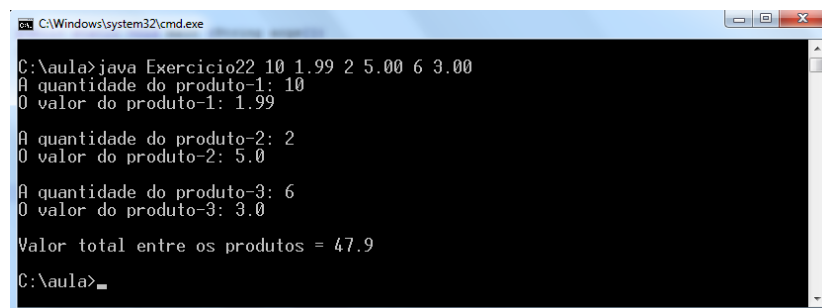
#### Atividades - 4

1) Crie uma classe que receba três nomes qualquer por meio da linha de execução do programa (prompt) e os imprima na tela da seguinte maneira: o primeiro e o último nome serão impressos na primeira linha um após o outro e o outro nome (o segundo) será impresso na segunda linha, conforme demonstra a figura.



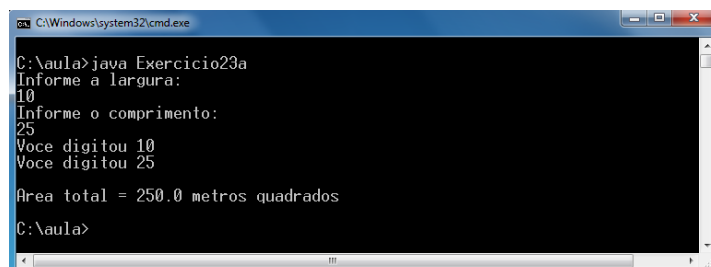
```
C:\Windows\system32\cmd.exe
C:\aula>java Exercicio21 Robson Aparecido Gomes
Robson Gomes
Aparecido
C:\aula>
```

2) Faça uma classe que receba por meio da linha de execução do programa (prompt) a quantidade e o valor de três produtos no seguinte formato: Quantidade1 Valor1 Quantidade2 Valor2 Quantidade3 Valor3. O programa deve calcular esses valores seguindo a fórmula:  $\text{total} = \text{Quantidade1} \times \text{Valor1} + \text{Quantidade2} \times \text{Valor2} + \text{Quantidade3} \times \text{Valor3}$ . O valor total deve ser apresentado no final da execução do programa, conforme indica a figura.



```
C:\Windows\system32\cmd.exe
C:\aula>java Exercicio22 10 1.99 2 5.00 6 3.00
A quantidade do produto-1: 10
O valor do produto-1: 1.99
A quantidade do produto-2: 2
O valor do produto-2: 5.0
A quantidade do produto-3: 6
O valor do produto-3: 3.0
Valor total entre os produtos = 47.9
C:\aula>
```

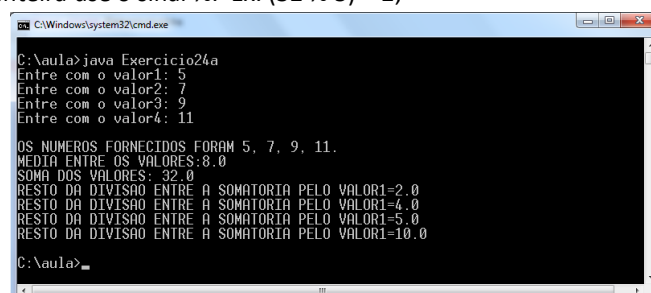
3) Crie uma classe que receba a largura e o comprimento de um lote de terra e mostre a área total existente, conforme a figura. Utilizando a seguinte fórmula:  $(\text{Área} = \text{Largura} \times \text{Comprimento})$  utilize a cláusula `import java.util.Scanner;`



```
C:\Windows\system32\cmd.exe
C:\aula>java Exercicio23a
Informe a largura:
10
Informe o comprimento:
25
Voce digitou 10
Voce digitou 25
Area total = 250.0 metros quadrados
C:\aula>
```

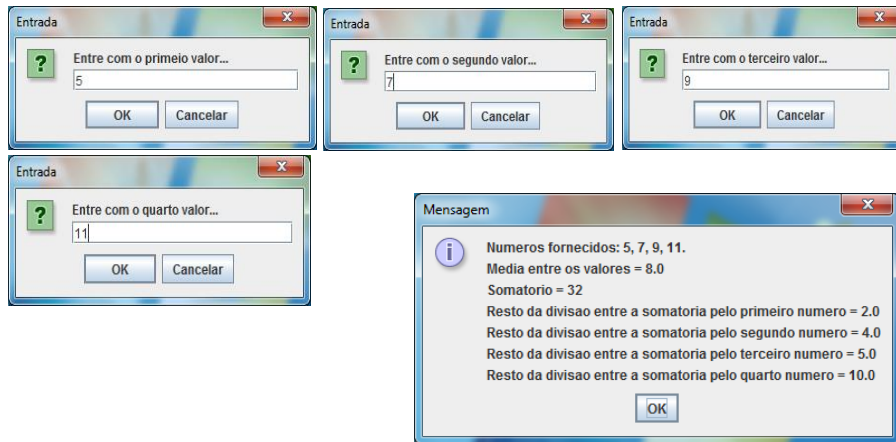
4) Crie uma classe que receba quatro valores qualquer e mostre a média e o somatório entre eles e o resto da divisão do somatório por cada um dos valores, conforme figura. Para quebrar uma linha na mensagem utilize `"\n"`.

Para saber o resto da divisão inteira use o sinal `%`. Ex.  $(32 \% 5) = 2$ ;

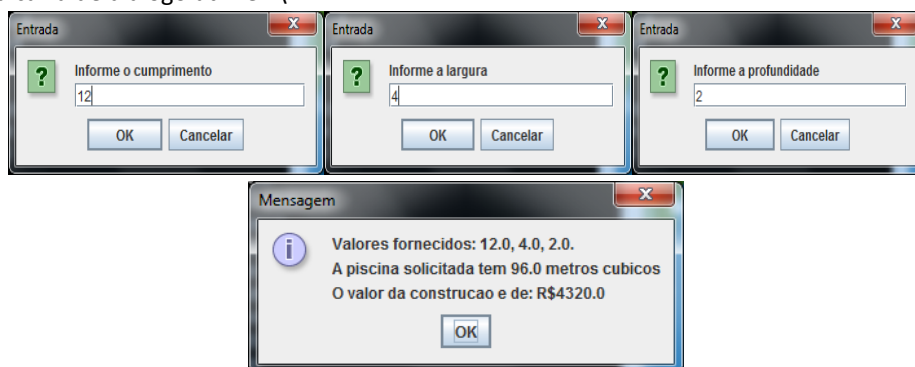


```
C:\Windows\system32\cmd.exe
C:\aula>java Exercicio24a
Entre com o valor1: 5
Entre com o valor2: 7
Entre com o valor3: 9
Entre com o valor4: 11
OS NUMEROS FORNECIDOS FORAM 5, 7, 9, 11.
MEDIA ENTRE OS VALORES:8.0
SOMA DOS VALORES: 32.0
RESTO DA DIVISAO ENTRE A SOMATORIA PELO VALOR1=2.0
RESTO DA DIVISAO ENTRE A SOMATORIA PELO VALOR2=4.0
RESTO DA DIVISAO ENTRE A SOMATORIA PELO VALOR3=5.0
RESTO DA DIVISAO ENTRE A SOMATORIA PELO VALOR4=10.0
C:\aula>
```

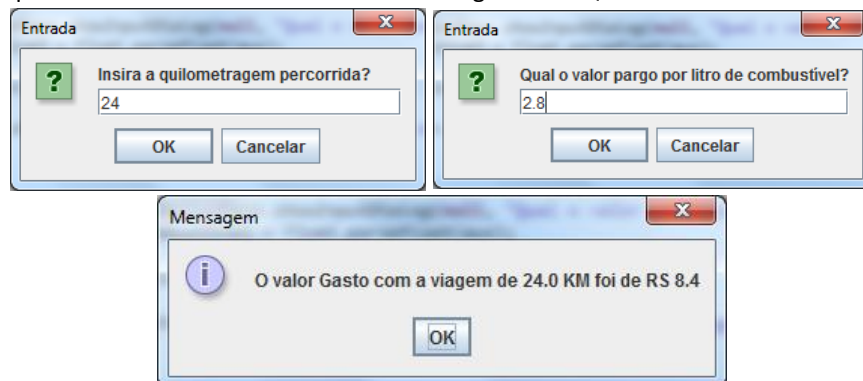
5) Crie uma classe que receba quatro valores qualquer e mostre a média e o somatório entre eles e o resto da divisão do somatório por cada um dos valores, conforme as figuras. Utilize desta vez a caixa de diálogo com a classe JOptionPane para a entrada e saída de dados. Para quebrar uma linha dentro da caixa de diálogo utilize “\n”.



6) Desenvolva um programa em que uma determinada pessoa que trabalha com piscinas precisa calcular o valor das construções de piscinas solicitadas pelos clientes, sendo que os clientes sempre fornecem o comprimento, largura e a profundidade da piscina a ser construída. A pessoa que constrói as piscinas cobra valor de R\$ 45 por m3. Utilize a caixa de diálogo com a classe JOptionPane para a entrada e saída de dados. Para quebrar uma linha dentro da caixa de diálogo utilize “\n”.



7) Desenvolva um programa que leia a quilometragem percorrida por um determinado veículo e leia o valor do combustível por litro. Este programa deverá calcular o gasto do veículo, sabendo que o veículo percorre 8Km/litro, conforme as figuras abaixo. Utilize a caixa de diálogo com a classe JOptionPane para a entrada e saída de dados. Para quebrar uma linha dentro da caixa de diálogo utilize “\n”.



8) Desenvolva um programa para ler o raio e a altura de uma lata de óleo e calcular o valor do seu volume, utilizando a fórmula:  $VOLUME = 3.14159 * R * R * ALTURA$ .

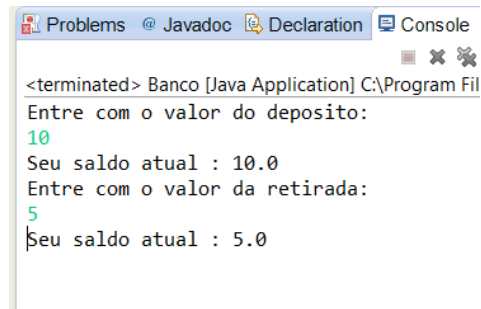
**9)** Desenvolva um programa para ler o preço de um produto e calcular o preço à vista deste produto com 10% de desconto. O programa deverá realizar os seguintes passos: Estabelecer a leitura da variável PRECO (preço bruto do produto);

Calcular o valor do desconto (DESCONTO) dividindo o preço por 100 e multiplicando pelo percentual de desconto  $((PRECO/100)*10)$ .

Calcular o preço do produto à vista (PRECO\_VISTA), descontando o valor do desconto do preço bruto do produto  $(PRECO - DESCONTO)$ .

Apresentar os valores do preço bruto e do preço à vista, além do desconto: PRECO, DESCONTO e PRECO\_VISTA.

**10)** Usando a classe **Scanner** para entrada de dados, faça uma classe que simule uma conta bancária cujo saldo inicial seja zero, ou seja as variáveis **saldo** **deposito** **retirada** inicializadas com **zero**. A seguir, simule um depósito num valor qualquer e mostre o saldo atual. Depois disso simule uma retirada (débito) qualquer e apresente o saldo final.



```
Problems  @ Javadoc  Declaration  Console
<terminated> Banco [Java Application] C:\Program Fil
Entre com o valor do deposito:
10
Seu saldo atual : 10.0
Entre com o valor da retirada:
5
Seu saldo atual : 5.0
```