

## MÓDULO 02 – PRÁTICA DE LABORATÓRIO I

### 1 INTRODUÇÃO

Nos tópicos a seguir **estaremos praticando os conceitos vistos no módulo 2 de Programação I**. Revise estes conceitos no capítulo do livro de apoio de Programação I:

- Capítulo 02 - ELEMENTOS BÁSICOS;
- Capítulo 03 - OS PRIMEIROS COMANDOS.

Na sequência, siga as instruções deste documento e tire suas dúvidas com o tutor no fórum de dúvidas.

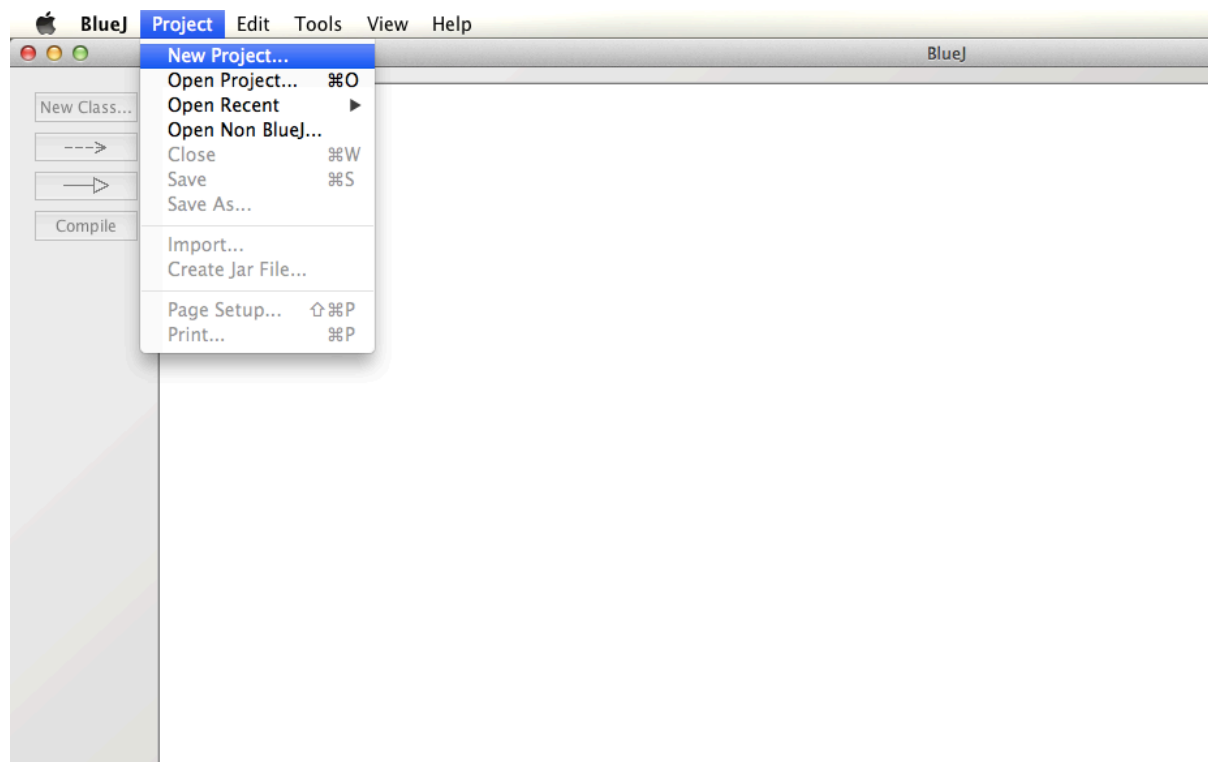
### 2 ELEMENTOS BÁSICOS E PRIMEIROS COMANDOS EM JAVA

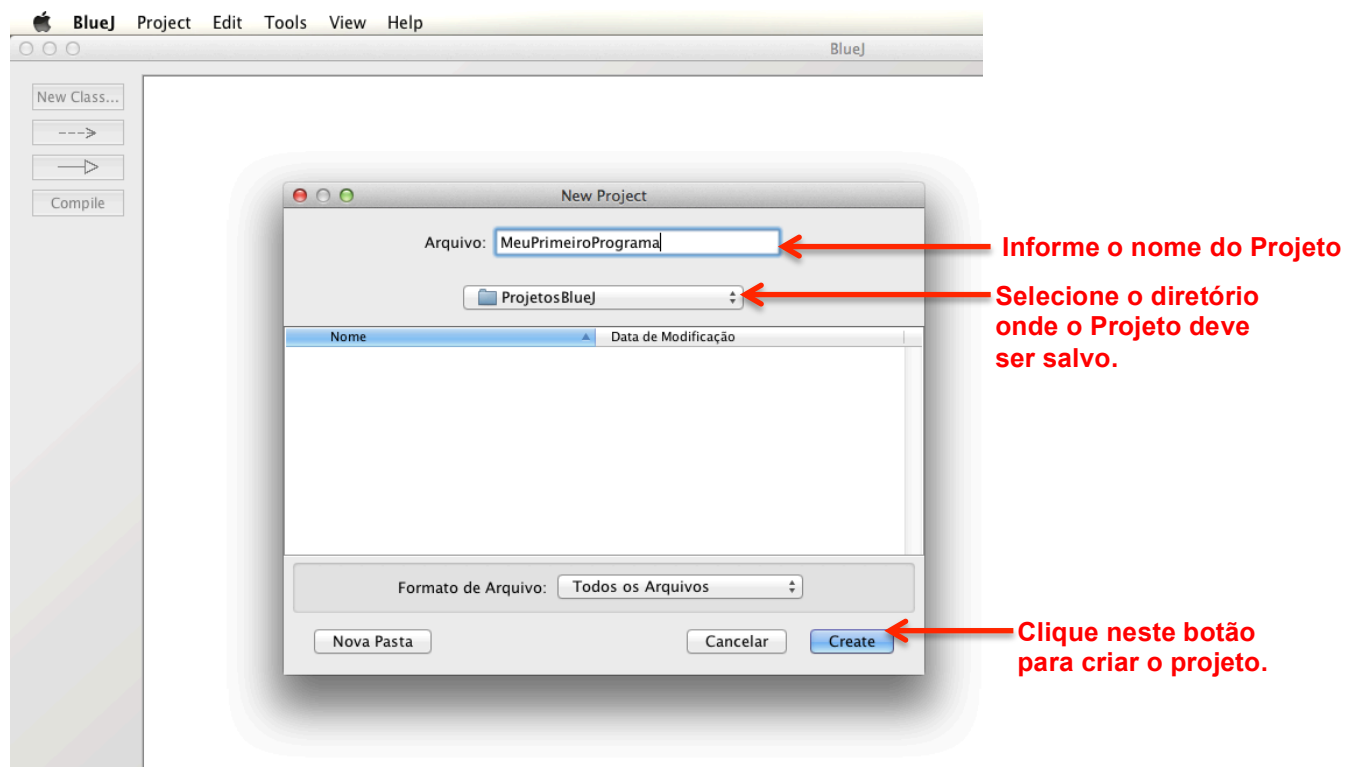
Como vimos no livro de apoio de Programação I, a linguagem Java possui alguns elementos básicos que irá permitir o desenvolvimento de programas. Neste tópico, estaremos praticando os elementos básicos e os primeiros comando em Java apresentados nos Capítulos 02 e 03 do livro de Programação I.

#### 2.1 PRIMEIRO EXEMPLO COM COMANDO DE SAÍDA

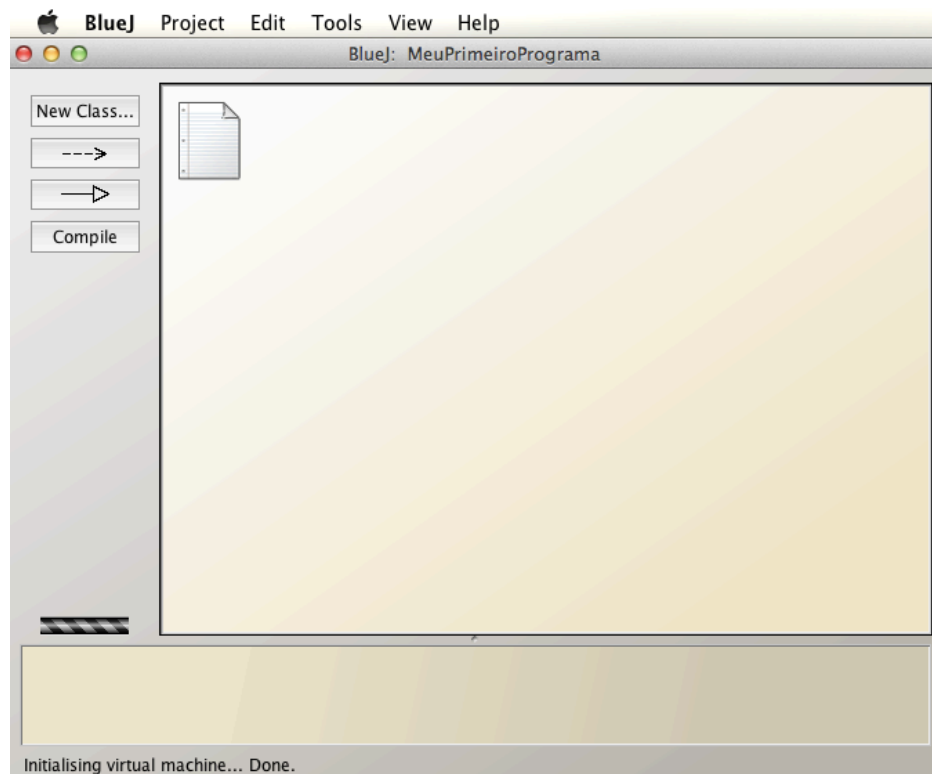
Vamos iniciar a prática verificando como utilizar o comando de saída de dados em Java. Por meio deste comando, será possível apresentar dados e informações na tela do BlueJ.

Abra o BlueJ e crie um novo projeto com o nome: MeuPrimeiroPrograma, conforme ilustra as imagens abaixo:



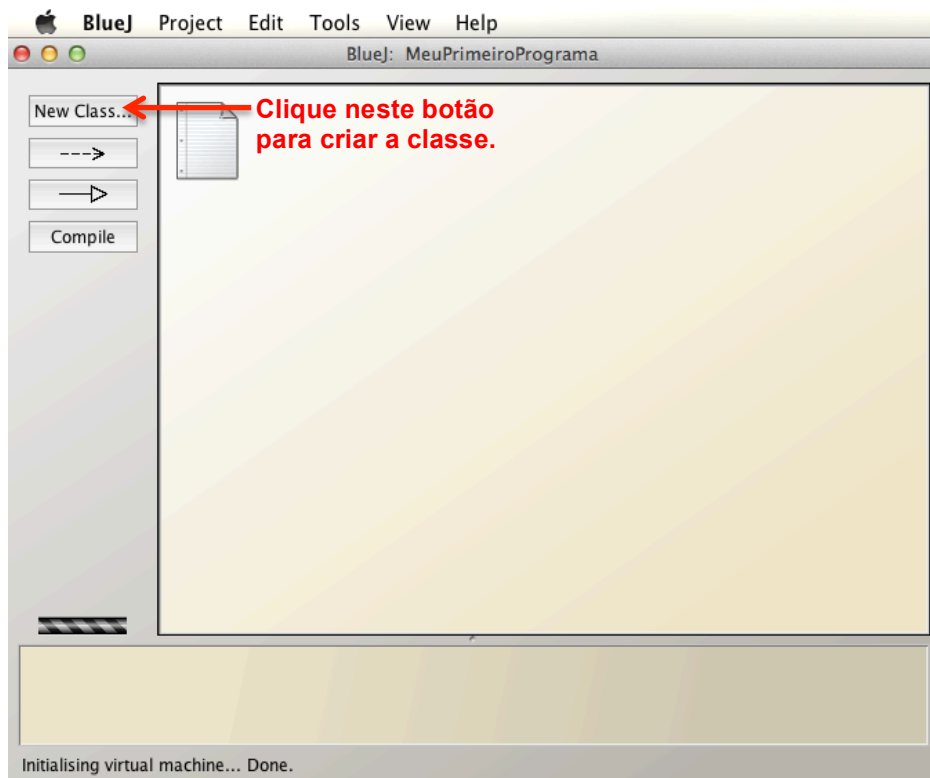


Após informar o nome do Projeto e selecionar o diretório destino onde o Projeto será salvo, clique no botão: Create para criar o Projeto. Ao criar o projeto, a tela abaixo será apresentada:

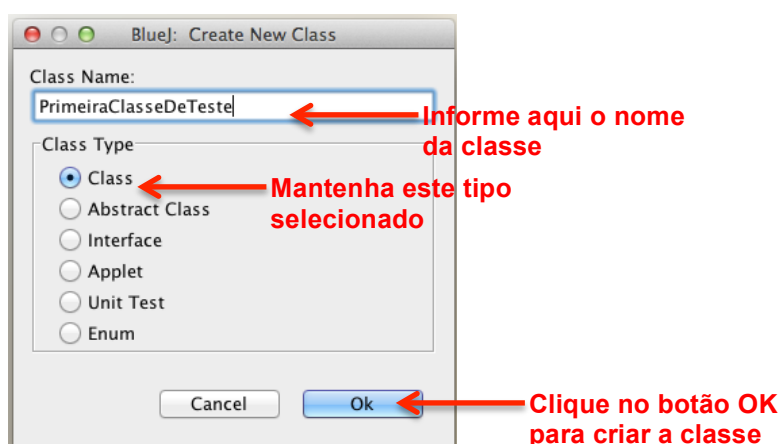


Vamos agora criar uma classe de teste. Por meio desta classe conseguiremos praticar os conceitos e realizar os exercícios de Laboratório I. **Atenção:** Você verá mais detalhes sobre a classe de teste nos próximos módulos de Programação e Laboratório I, então neste momento, não se preocupe com a estrutura da classe de teste.

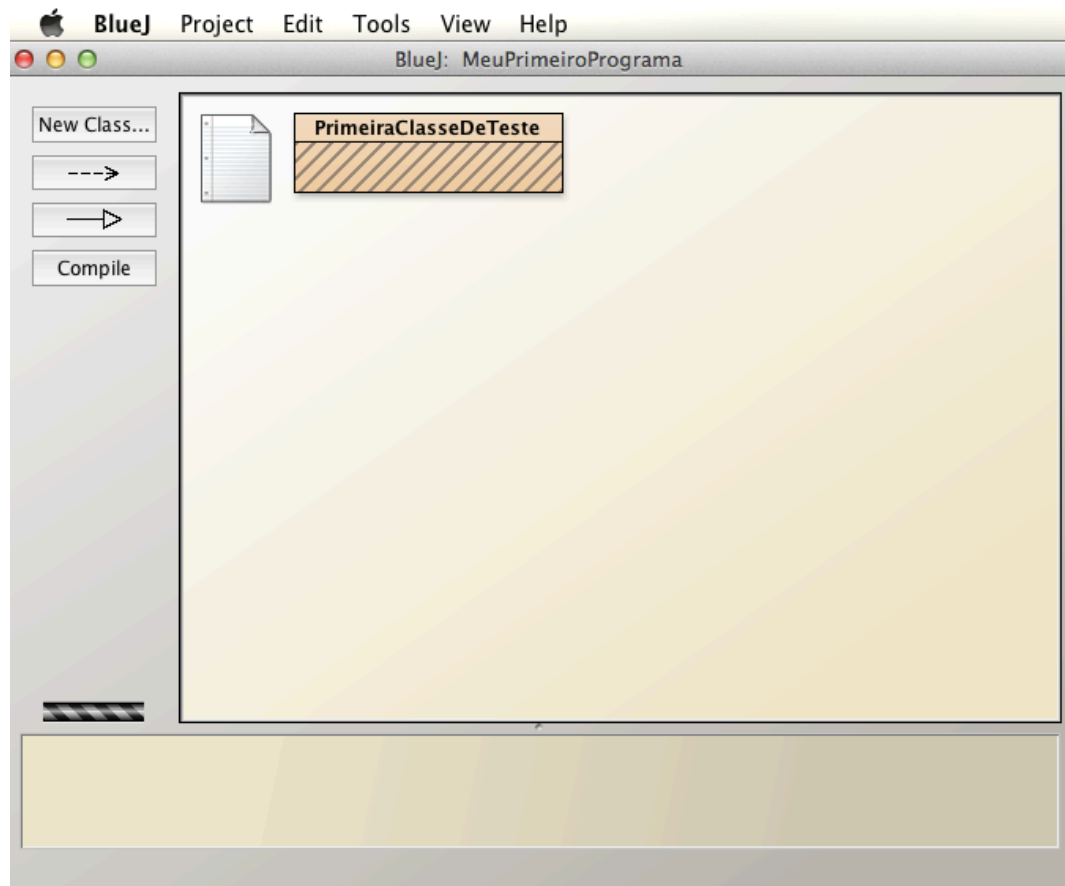
Para criar a classe, siga os passos a seguir. Clique no botão: New Class:



Será apresentada a seguinte tela para colocarmos o nome da classe:



Após a criação da classe, o BlueJ irá apresentar a classe criada conforme a tela abaixo:



A seguir, estaremos editando a classe de teste e utilizando o comando de saída. Para isso, precisamos primeiro verificar qual é a estrutura da classe de teste. A estrutura da classe de teste é apresentada na imagem abaixo:

```
/*
 * Não se preocupe com esta estrutura neste momento
 * você estará estudando mais sobre a classe de teste
 * nos próximos módulos
 */
public class NomeDaClasseDeTeste
{
    public static void main(String args[])
    {
        /*
         * Aqui neste espaço estaremos praticando os conceitos
         */
    }
}
```

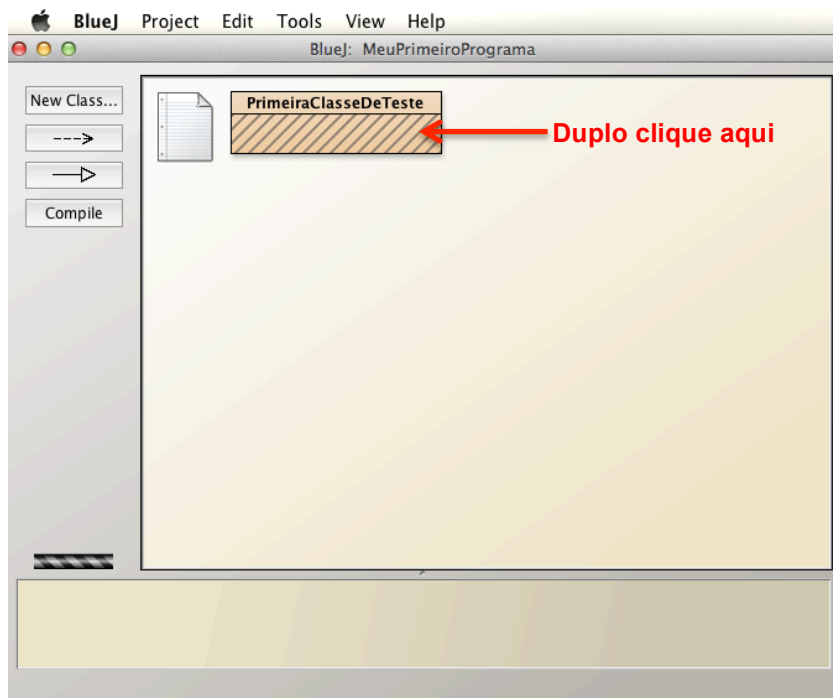
← **Modificaremos apenas este nome**

← **E este conteúdo. Aqui praticaremos os conceitos.**

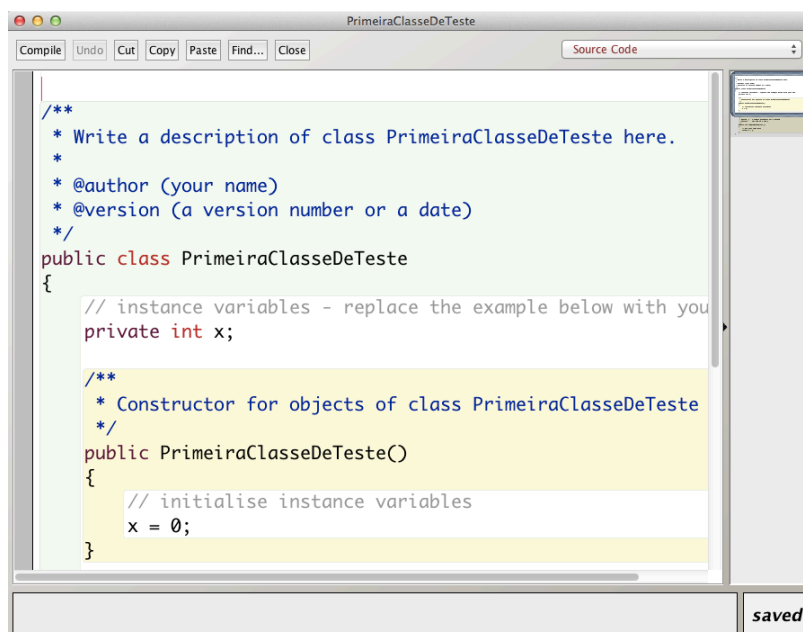
Não se preocupe com esta estrutura neste momento, pois estaremos estudando mais detalhadamente esta estrutura nos próximos módulos. O que precisamos saber neste momento é que sempre utilizaremos esta mesma estrutura para praticar os conceitos. Os únicos itens que estaremos modificando nesta estrutura, durante esta nossa prática de Laboratório I, é o nome da

classe, que na imagem está como: **NomeDaClasseDeTeste**, e o conteúdo que estará sendo praticado, que será incluído onde está o comentário: “Aqui neste espaço estaremos praticando os conceitos”.

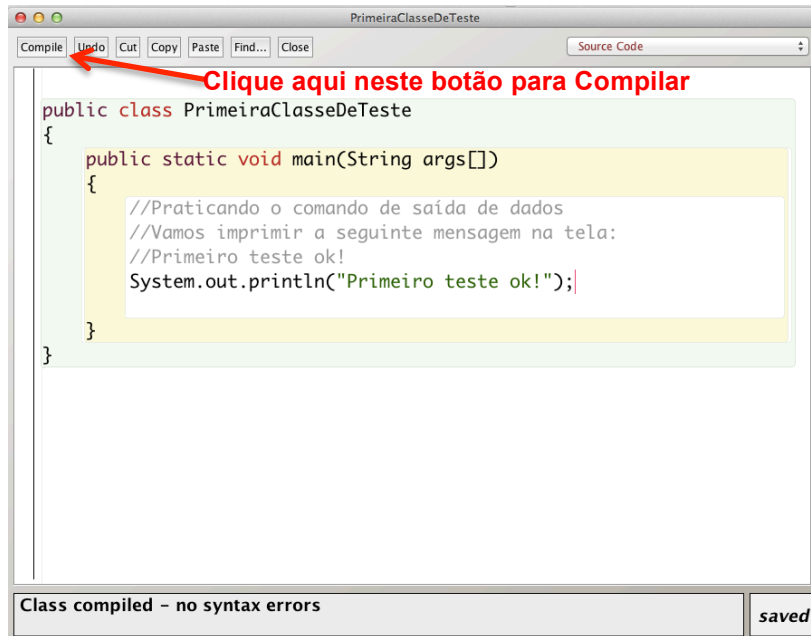
Vamos então editar a classe que criamos, a classe: PrimeiraClasseDeTeste. Para isso, dê duplo clique no ícone da classe, conforme ilustra a imagem abaixo:



Será aberto a tela de edição conforme ilustra a imagem abaixo. Como você pode verificar, o BlueJ já apresenta, por padrão, um conteúdo automático para auxiliar na criação das classes. Neste momento não estaremos utilizando esta estrutura, então pode selecionar todo o conteúdo da classe e apagar.



Na sequência, vamos preparar a nossa classe com a estrutura padrão da classe de teste e utilizar o comando de saída para apresentar uma mensagem na tela. Para isso, altere a sua classe para que fique como a imagem abaixo. **Atenção:** Para que você possa comparar a sua classe, com a classe abaixo, faça o download deste projeto que está disponível para download no módulo 01.

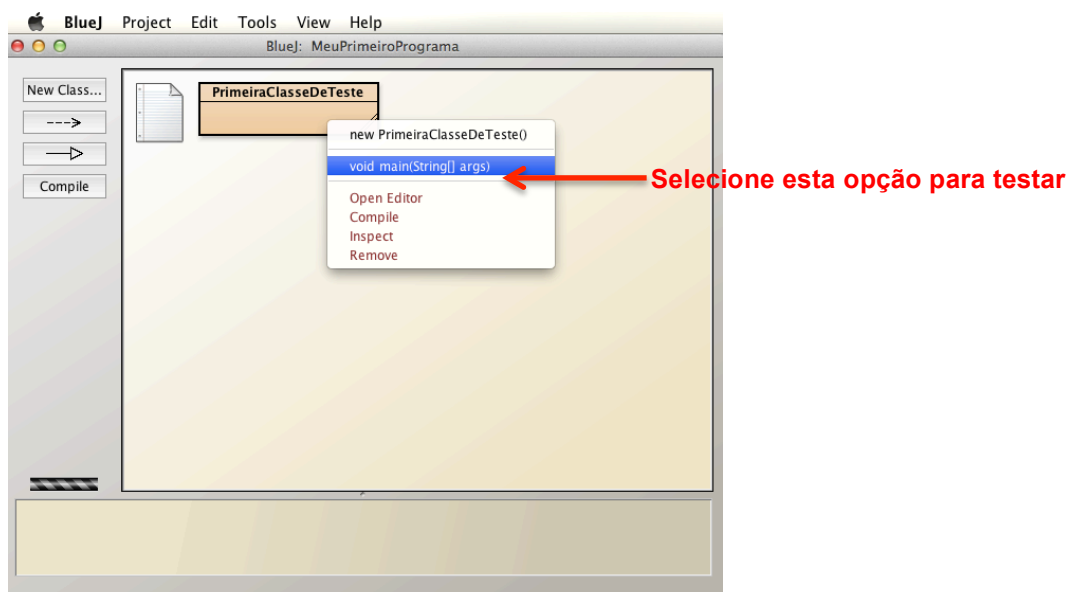


Depois de editar a classe, é necessário compilar. Para isso, clique no botão: [Compile]. Como vimos no livro de apoio de **Programação I**, no momento da compilação, o compilador do Java fará a verificação da sintaxe e se tudo estiver ok, a classe será compilada e o arquivo: .class será criado.

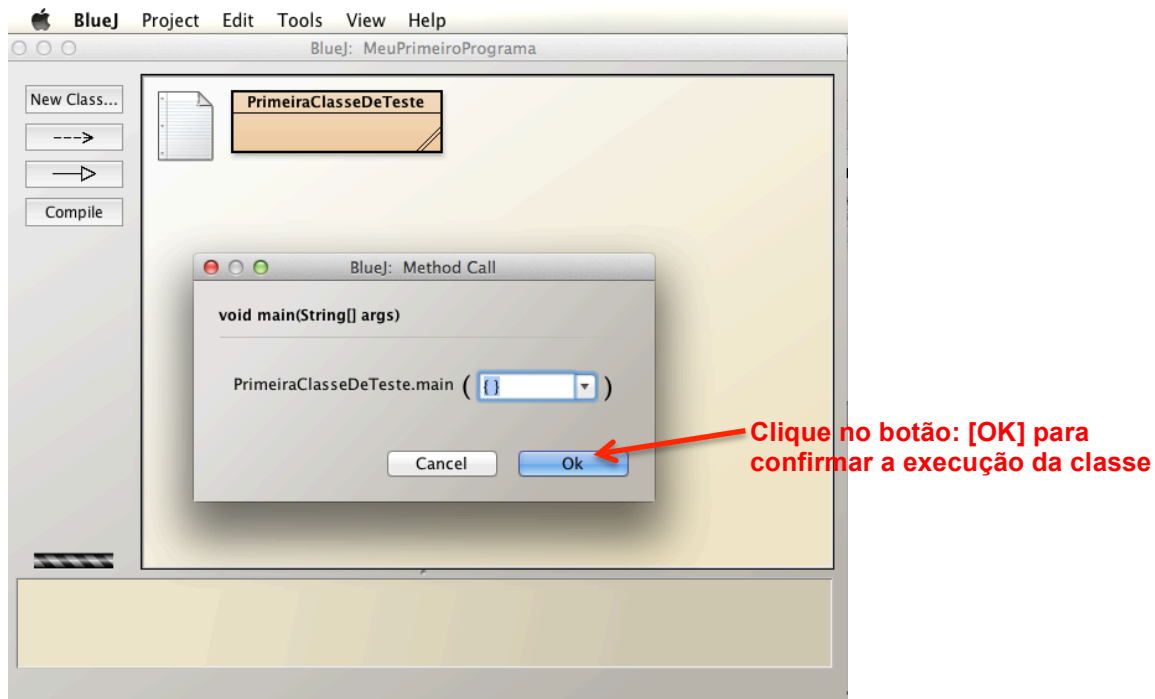
Com a classe compilada, podemos fechar a tela de edição, por meio do botão: [Close] e testar o nosso exemplo. Para testar precisaremos executar as seguintes ações no BlueJ:

- Clicar uma vez em cima do ícone da classe;
- E selecionar a opção: void main (String[] args).

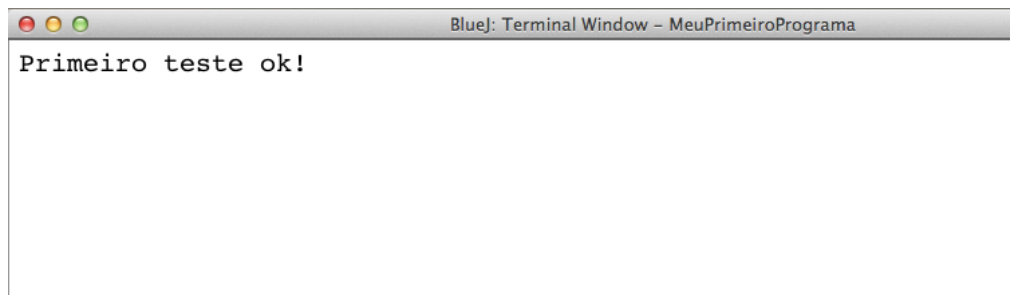
Na imagem abaixo é apresentada as ações que devemos executar:



Depois clique no botão: [OK] da próxima tela que será apresentada, para confirmar a execução da classe, conforme imagem abaixo:



Será apresenta a tela (Console do BlueJ) com a execução e o resultado do nosso comando de saída:



## 2.2 SEGUNDO EXEMPLO COM OS COMANDOS DE SAÍDA

Agora que já vimos como praticar os comandos no BlueJ, vamos praticar mais alguns exemplos com os comandos de saída. Para isso, vamos lembrar quais os três (3) tipos de comandos de saída que temos:

- `System.out.print (<aqui os itens a serem exibidos na tela>)` = apresenta os itens a serem exibidos na tela e mantém o cursor ao final do item exibido;
- `System.out.println (<aqui os itens a serem exibidos na tela>)` = apresenta os itens a serem exibidos na tela e coloca o cursor na próxima linha;
- `System.out.printf (<"formatação"><aqui os itens a serem exibidos na tela>)` = apresenta os itens a serem exibidos de acordo com a formatação definida. Podemos utilizar os seguintes códigos de formatação:

- %d – será substituído por um valor inteiro (valores inteiros);
- %f – será substituído por um valor real (%8.2f reservando 8 posições da tela, das quais 2 serão usadas para as casas decimais);
- %c – será substituído por um caractere;
- %s – será substituído por uma cadeia de caracteres;
- %b – será substituído por um valor lógico (boolean);
- \n – quebra a linha.

Vamos agora praticar estes comandos no BlueJ. Para isso vamos utilizar os seguintes exemplos:

1. Imprimir a seguinte mensagem:

“Seja bem-vindo ao...”  
“...Mundo Java!”

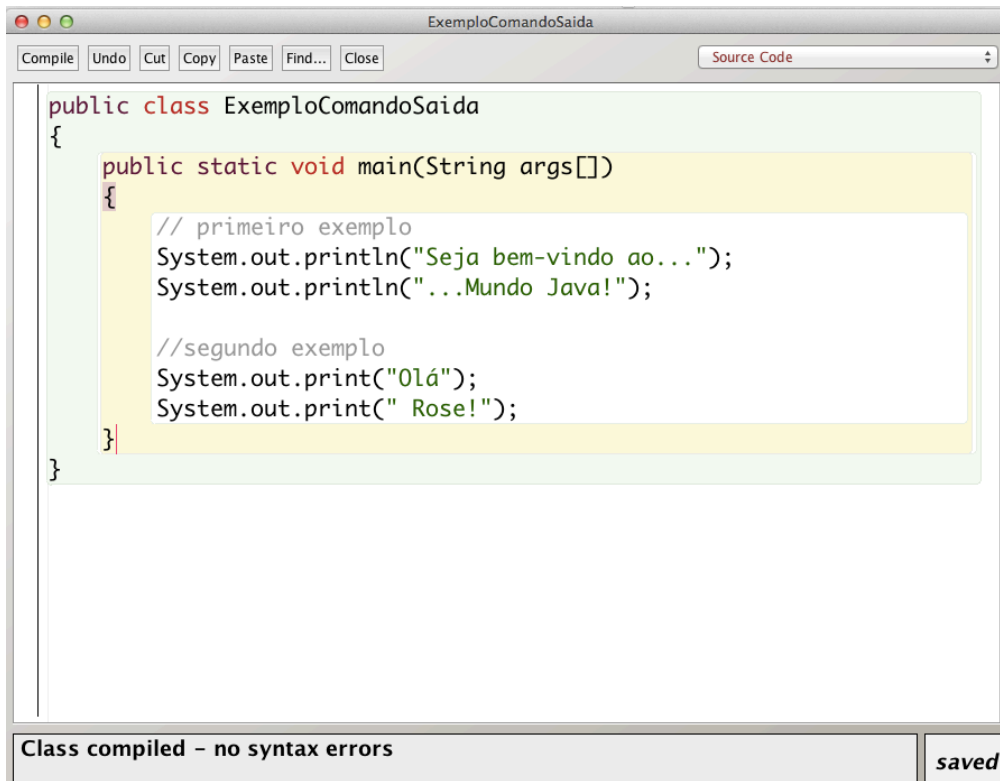
2. Imprimir a seguinte mensagem:

“Olá” “[seuNome]!”

Para o primeiro exemplo, temos duas mensagens para apresentar em linhas separadas. Sabemos que o comando: **System.out.println** apresenta o item na tela e pula uma linha. Então, podemos usar este comando para apresentar as duas mensagens.

Para o segundo exemplo, temos duas mensagens para apresentar na mesma linha. Sabemos que o comando: **System.out.print** apresenta o item na tela e continua na mesma linha. Então, podemos usar este comando para apresentar as duas mensagens.

Vamos verificar como ficaria uma classe de teste para praticar estes dois exemplos:



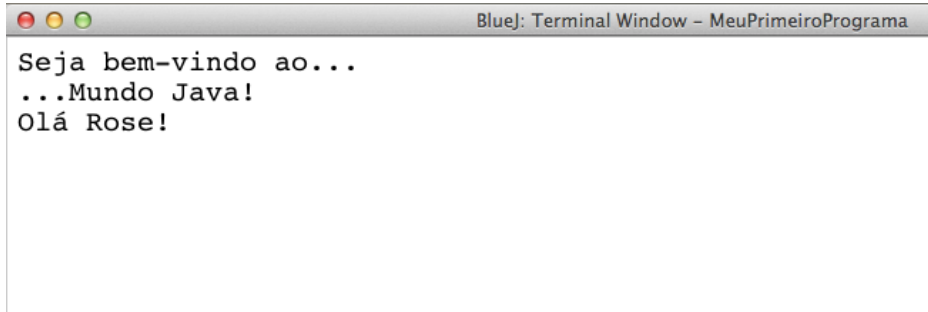
```
public class ExemploComandoSaida
{
    public static void main(String args[])
    {
        // primeiro exemplo
        System.out.println("Seja bem-vindo ao...");
        System.out.println("...Mundo Java!");

        //segundo exemplo
        System.out.print("Olá");
        System.out.print(" Rose!");
    }
}
```

Class compiled - no syntax errors saved



Depois de compilar a classe, vamos testá-la:



```
BlueJ: Terminal Window - MeuPrimeiroPrograma
Seja bem-vindo ao...
...Mundo Java!
Olá Rose!
```

Estaremos praticando o comando: **System.out.printf** nos próximos exemplos, pois estaremos utilizando os tipos de dados para verificar como utilizar a formatação.

**Atenção:** Para que você possa comparar a sua classe, com a classe desenvolvida neste exemplo, faça o download deste projeto que está disponível para download no módulo 01.

## 2.3 TERCEIRO EXEMPLO COM O COMANDO DE ENTRADA DE DADOS

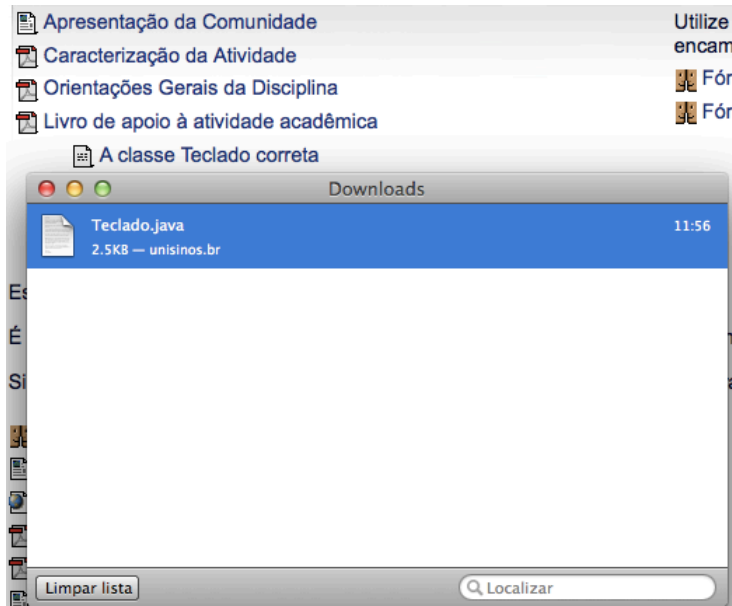
Neste tópico vamos praticar os comandos de: **Entrada de Dados**. Para utilizar os comandos de entrada de dados, estaremos utilizando a classe **Teclado** disponibilizada no início das comunidades de Programação e Laboratório I. **Atenção:** Para verificar mais detalhes sobre a classe Teclado, revise o Capítulo 03 do livro de apoio de Programação I.

Antes de utilizarmos a classe Teclado, é importante incluirmos esta classe dentro do nosso Projeto. **Importante:** O procedimento que veremos a seguir deve ser realizado sempre que você fizer uso de comandos de entrada de dados e uso da classe Teclado.

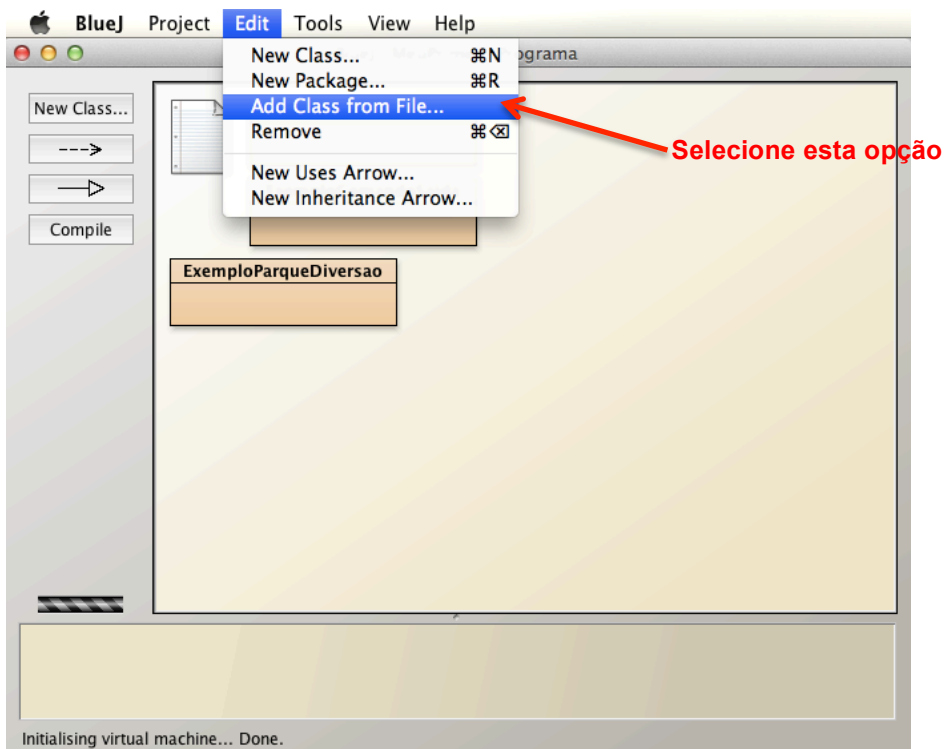
Primeiro, faça o download da classe teclado para o seu computador. Você deve salvar o arquivo com o nome e extensão: **Teclado.java**. A figura abaixo ilustra como fazer o download e salvar o arquivo da classe Teclado no seu computador:



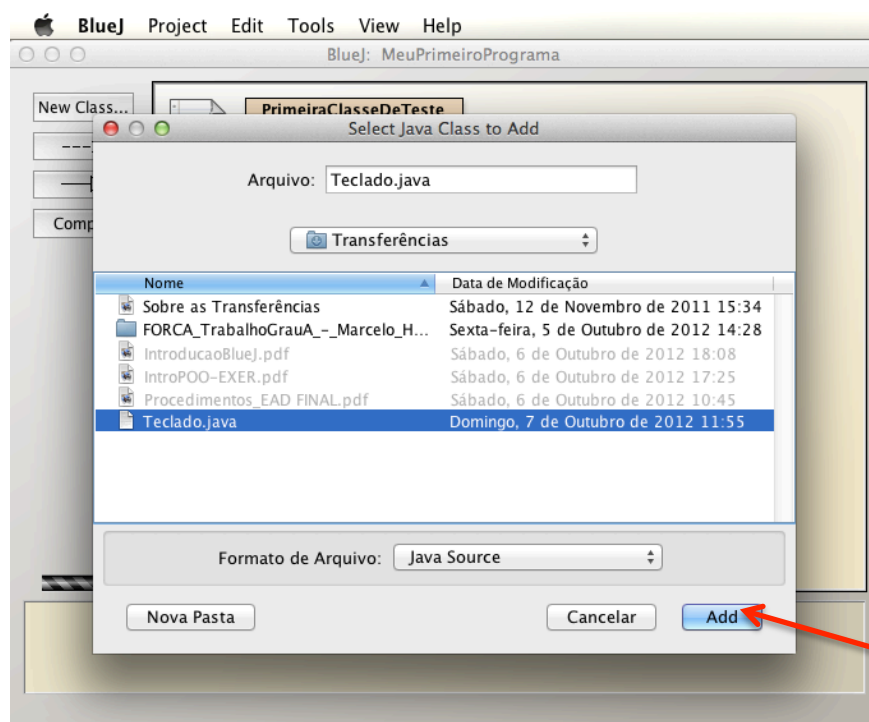
Será salvo em seu computador o arquivo: Teclado.java:



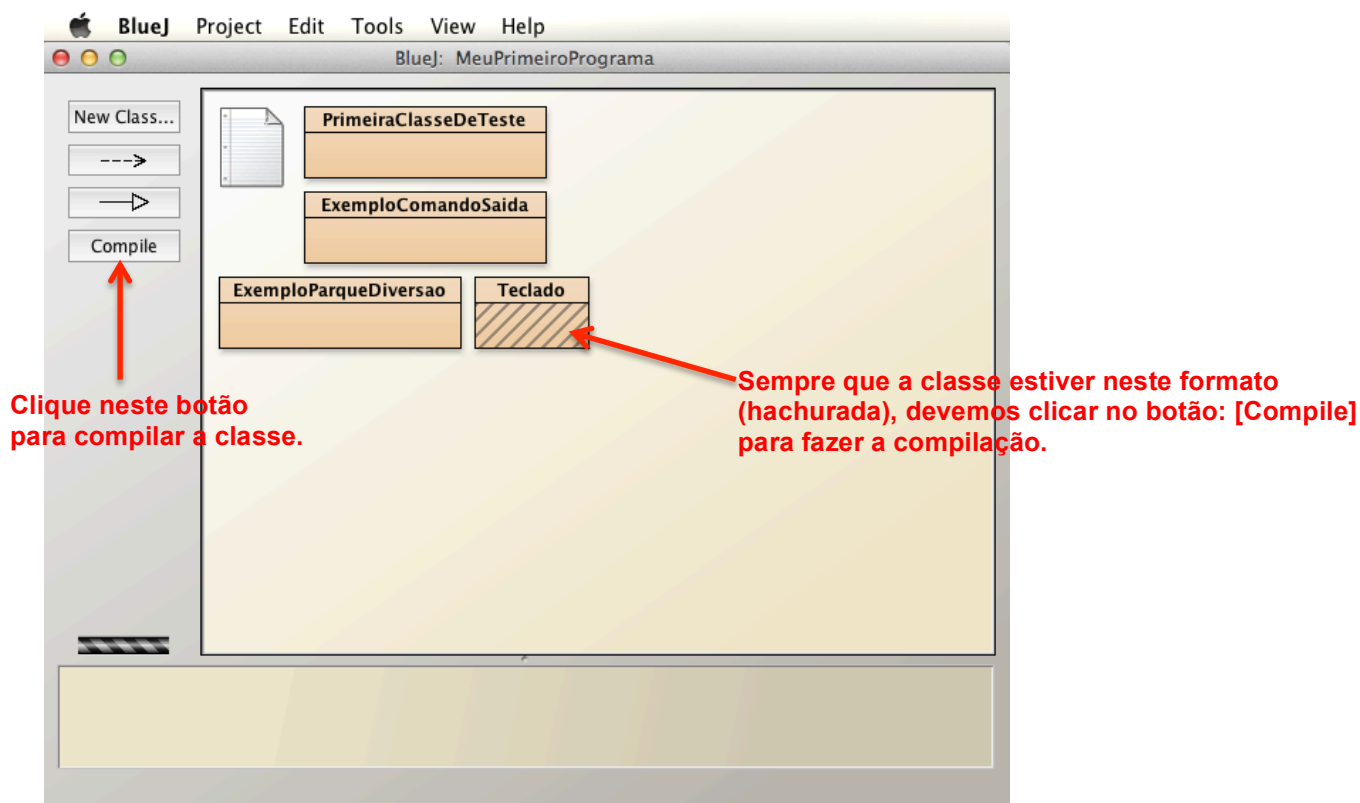
Com o projeto do BlueJ aberto, clique no menu: **Edit** e selecione a opção: **Add class From File...** conforme ilustra a imagem abaixo:



Será aberto uma janela para que você localize o arquivo: Teclado.java que você fez download. Localize o arquivo e clique no botão: [Add] conforme ilustra a imagem abaixo:



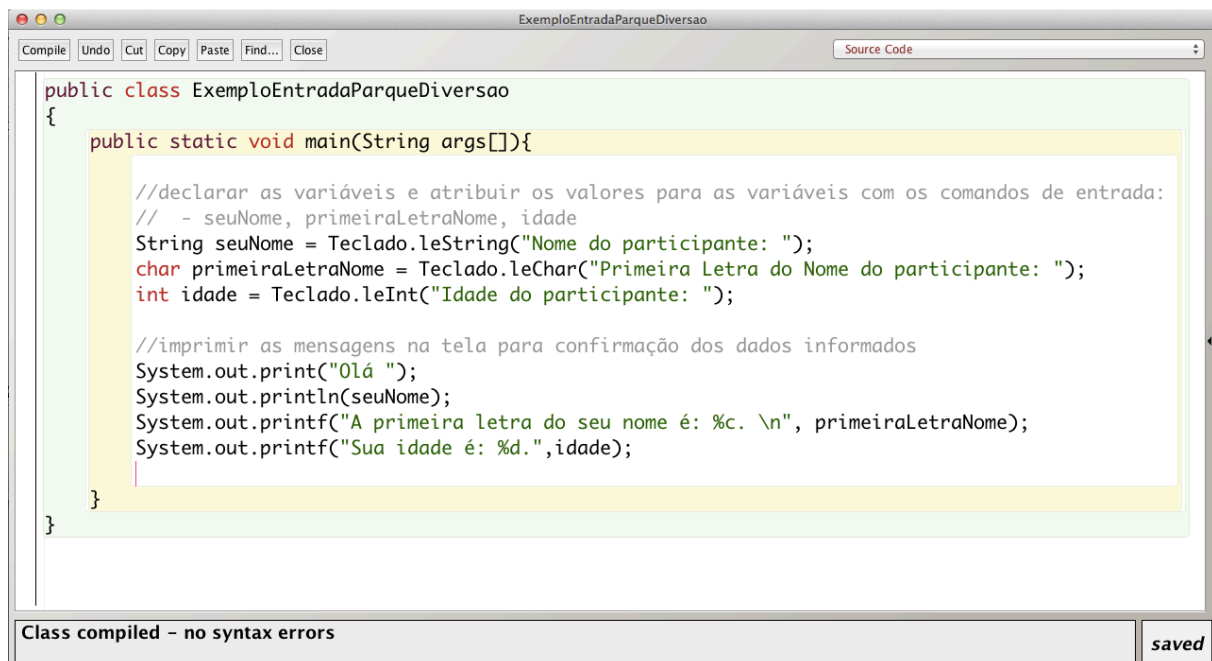
A classe Teclado.java será adicionada ao projeto e poderemos utilizá-la para praticar os comandos de entrada de dados. Antes porém de utilizarmos a classe Teclado, teremos que compilar a mesma. Para isso clique no botão: [Compile] para compilar, conforme ilustra a imagem abaixo:



Pronto! Agora já podemos utilizar a classe Teclado para praticar os comandos de entrada.

Voltando a situação do parque de diversão. Vamos imaginar que na entrada do Parque, o atendente pergunta os seus dados básicos para registrar no sistema que emite o cartão que será utilizado para acesso aos brinquedos. Para que o atendente possa registrar os dados que você informou a ele, utilizaremos os comandos de entrada de dados.

Vamos então criar uma classe de teste chamada: `ExemploEntradaParqueDiversao` conforme a imagem abaixo:



```

public class ExemploEntradaParqueDiversao
{
    public static void main(String args[]){

        //declarar as variáveis e atribuir os valores para as variáveis com os comandos de entrada:
        // - seuNome, primeiraLetraNome, idade
        String seuNome = Teclado.leString("Nome do participante: ");
        char primeiraLetraNome = Teclado.leChar("Primeira Letra do Nome do participante: ");
        int idade = Teclado.leInt("Idade do participante: ");

        //imprimir as mensagens na tela para confirmação dos dados informados
        System.out.print("Olá ");
        System.out.println(seuNome);
        System.out.printf("A primeira letra do seu nome é: %c. \n", primeiraLetraNome);
        System.out.printf("Sua idade é: %d.", idade);

    }
}

```

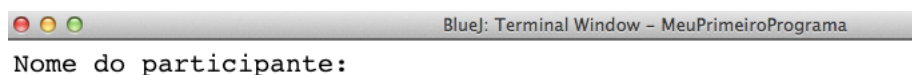
Class compiled - no syntax errors saved

Diferente do nosso exemplo anterior, o terceiro exemplo, os dados das variáveis foram atribuídos por meio dos comandos de entrada da classe: `Teclado`. Utilizamos os seguintes comandos:

- `Teclado.leString`("Mensagem que vai ser apresentada para o usuário na tela") = para ler dados do tipo de dado objeto: `String`;
- `Teclado.leChar`("Mensagem que vai ser apresentada para o usuário na tela") = para ler dados do tipo de dado primitivo: `char`;
- `Teclado.leInt`("Mensagem que vai ser apresentada para o usuário na tela") = para ler dados do tipo de dado primitivo: `int`.

Importante: Para cada tipo de dados existe um comando específico da classe: `Teclado` que devemos utilizar. O comando da classe `Teclado` deve estar sempre de acordo com o tipo de dados da variável que vai receber o valor.

Ao executarmos a classe de exemplo temos os seguintes resultados, conforme ilustra as imagens abaixo:



```

BlueJ: Terminal Window - MeuPrimeiroPrograma
Nome do participante:

```

BlueJ: Terminal Window - MeuPrimeiroPrograma

```
Nome do participante: Maria
Primeira Letra do Nome do participante:
```

BlueJ: Terminal Window - MeuPrimeiroPrograma

```
Nome do participante: Maria
Primeira Letra do Nome do participante: M
Idade do participante:
```

BlueJ: Terminal Window - MeuPrimeiroPrograma

```
Nome do participante: Maria
Primeira Letra do Nome do participante: M
Idade do participante: 15
Olá Maria
A primeira letra do seu nome é: M.
Sua idade é: 15.
```

**Atenção:** Para que você possa comparar a sua classe, com a classe desenvolvida neste exemplo, faça o download deste projeto que está disponível para download no módulo 01.