

1 Introdução

Esse roteiro trata do uso da ferramenta de desenvolvimento Java NetBeans. Essa ferramenta é classificada em uma categoria denominada de IDE (Integrated Development Environment), que traduzido corresponde a Ambiente Integrado de Desenvolvimento. Esse documento tem o objetivo de conduzir os iniciantes na ferramenta, mas não irá dar suporte a linguagem Java.

2 Introdução ao NetBeans

Iniciaremos pela apresentação da tela inicial da ferramenta. Quando o NetBeans é iniciado pela primeira vez, irá apresentar a tela de boas vindas, veja imagem a seguir:

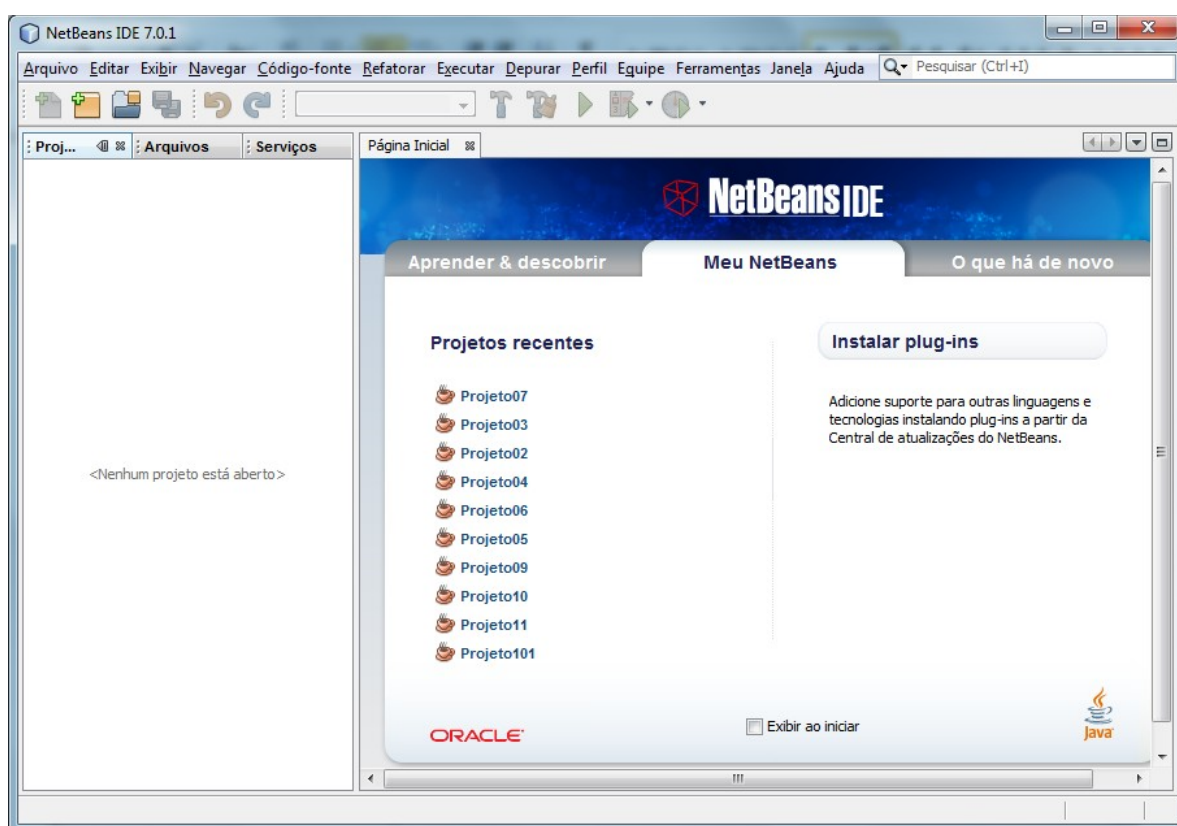


Figura 1. Tela inicial do NetBeans

Caso o computador que você está utilizando estiver conectado a Internet, o NetBeans irá atualizar essa tela com notícias, tutoriais e blogs. Também irá apresentar os projetos mais recentes desenvolvidos por essa instalação da IDE. Caso não queira que essa tela seja apresentada toda vez que o NetBeans for iniciado, vá ao final dela e desmarque a opção Exibir ao Iniciar. Caso deseje visualizar em algum momento, use o menu Ajuda, opção Página Iniciar, veja figura a seguir:

Roteiro de Uso da Ferramenta Netbeans

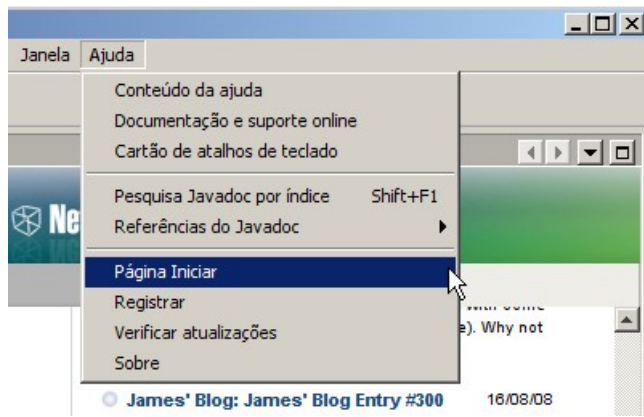
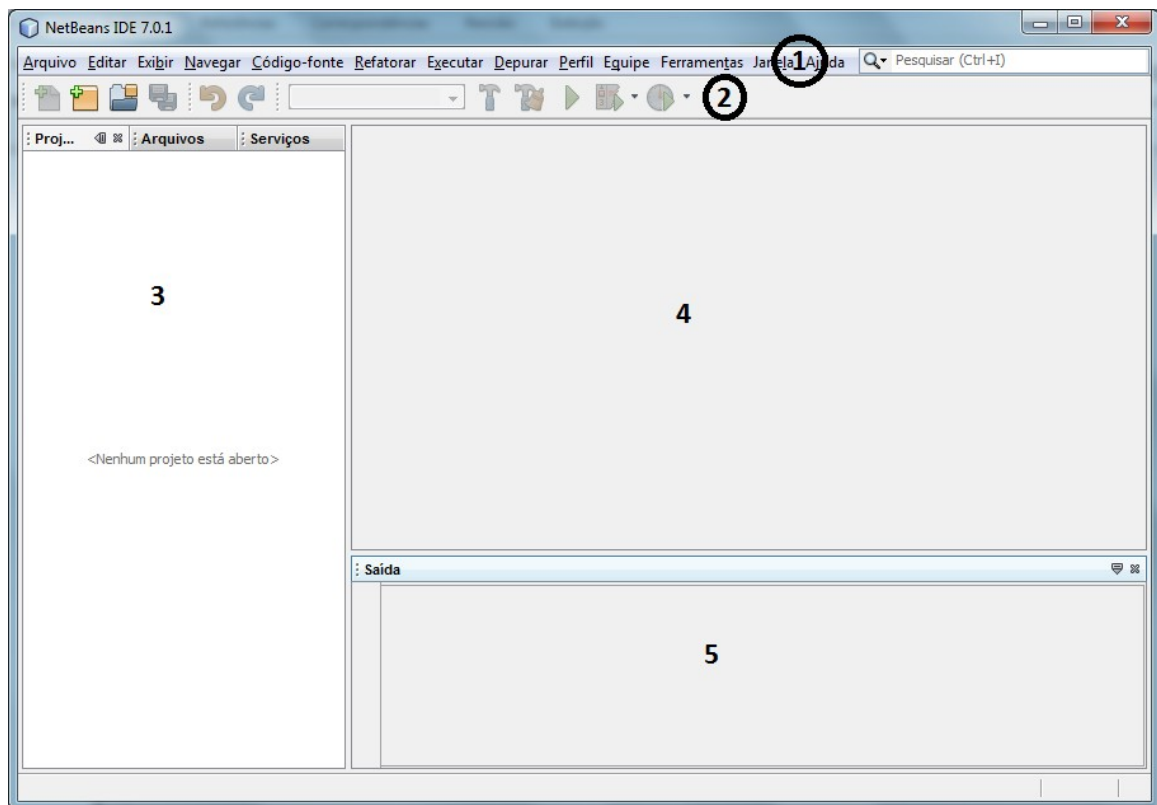


Figura 2. Visualizar Página Iniciar

Vamos agora analisarmos os principais elementos da IDE NetBeans, veja a imagem a seguir, com as indicações das regiões da tela destacadas.



O elemento indicado pelo número 1 é a barra de menus do NetBeans, que compreendem todas as funcionalidades de a ferramenta disponibiliza, gradativamente iremos conhecendo algumas delas. A indicação 2 aponta para a barra de botões, que são atalhos para as opções que estão presentes também no menu, contudo são aquelas mais usadas e por isso foram disponibilizadas no formato de botões. O elemento 3 é uma região que é usada para exibir informações do projeto em diferentes visualizações: visão de projeto, visão de pastas / arquivos e visão dos Serviços que o

Roteiro de Uso da Ferramenta Netbeans

projeto usa, este ultimo não será explorado nesse material, visto que esse conteúdo será abordado por disciplinas mais adiante do curso, tais como: conexões com banco de dados, contêiner web, servidores de aplicação etc. O elemento 4 é a área de edição de código fonte e outros elementos e por último o elemento 5 é a região destinada a saída de informações decorrentes da compilação e execução do projeto.

2.1 Como criar um novo projeto Java no NetBeans

O NetBeans, assim como a maioria das IDE's profissionais trabalham em um formato de projeto, assim, mesmo que a aplicação seja constituída de apenas uma classe Java, será necessário criar um projeto. O formato de projeto é interessante porque facilita a construção de aplicações que envolvem várias classes, já que, quem administra essas classes é a IDE.

Para criarmos um projeto novo no NetBeans podemos usar a opção de menu Arquivo / Novo Projeto, em seguida será apresentada a tela a seguir, veja imagem:

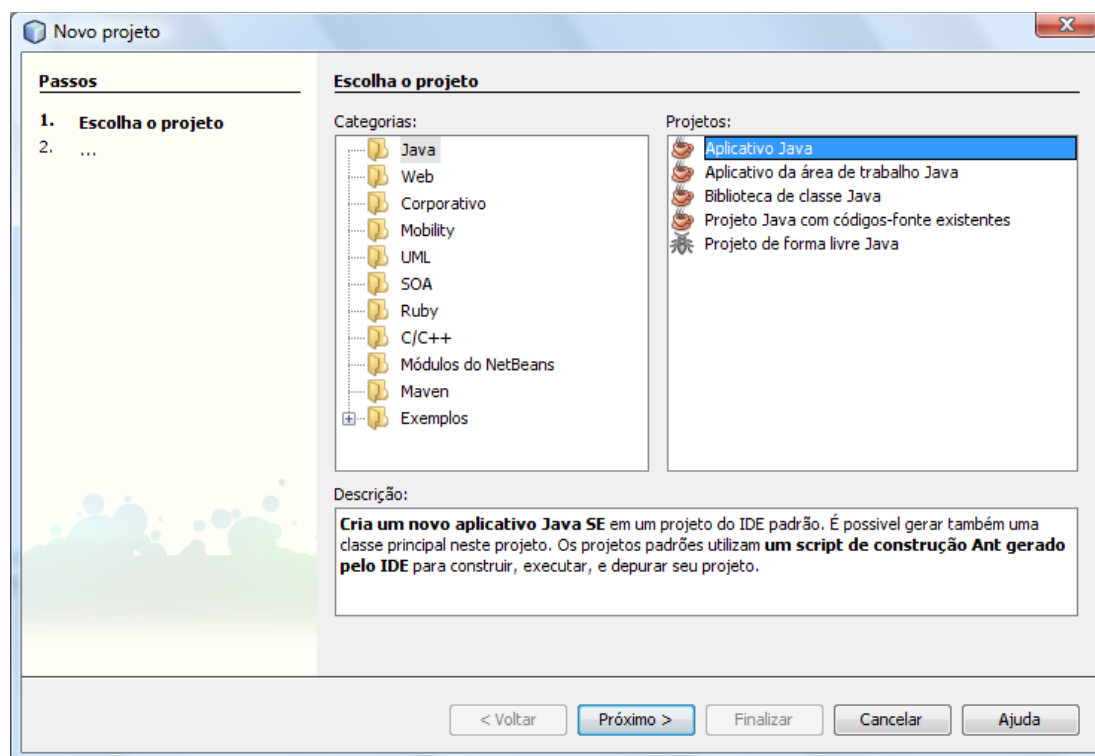


Figura 1. Novo Projeto Java

O segundo passo é escolher a categoria do projeto a ser criando, no nosso caso usaremos a categoria Java e o tipo de projeto Aplicativo Java. Clique no botão Próximo e será apresentada a tela a seguir:

Roteiro de Uso da Ferramenta Netbeans

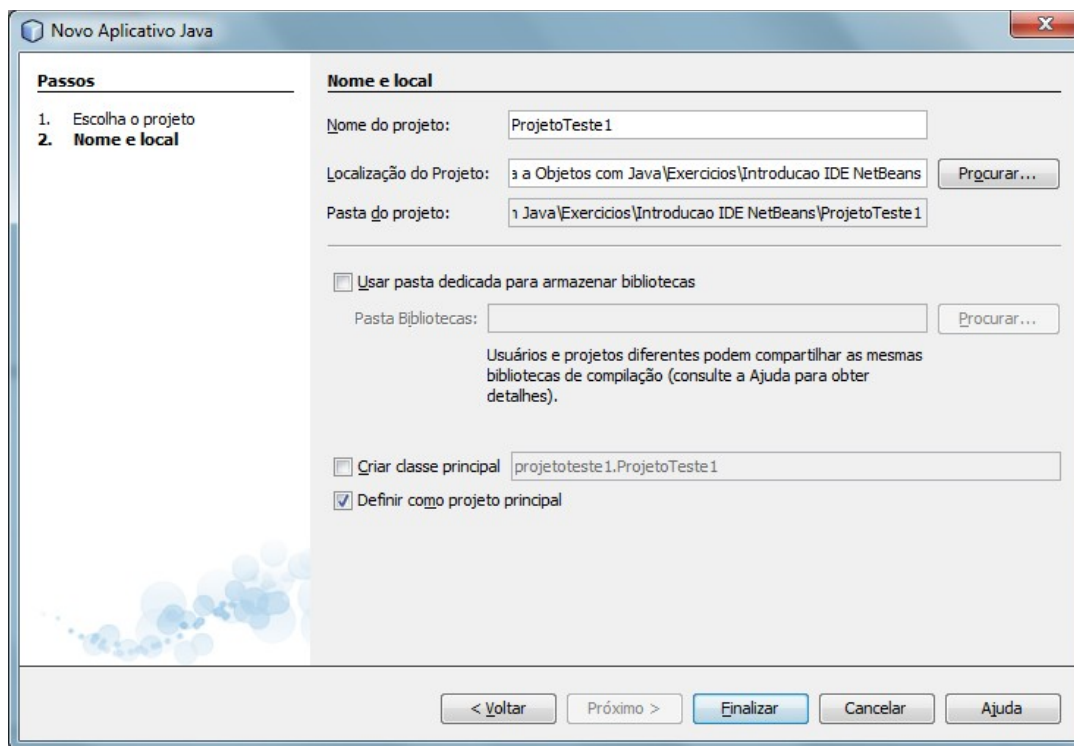
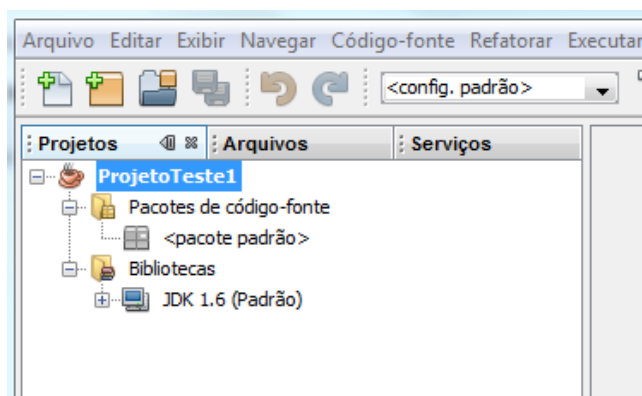


Figura 2. Definição de Características do Novo Projeto

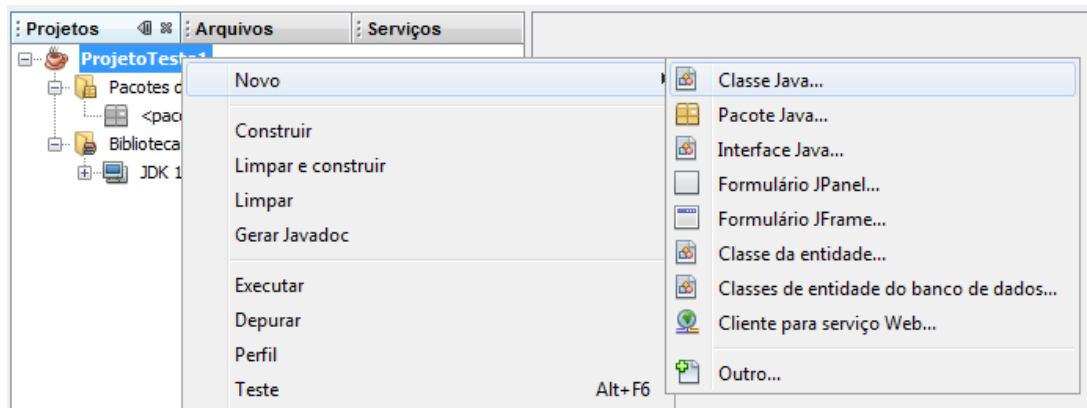
Na tela anterior devemos informar o nome do projeto a ser criado (ProjetoRetangulo), a localização do projeto, ou seja, onde o projeto será armazenado. O NetBeans irá sugerir a pasta NetBeansProjects que é uma pasta criada automaticamente para cada usuário que usar a IDE, mas você pode mudar isso por meio do botão Procurar. O campo pasta do projeto será gerado automaticamente em função do nome do projeto e da localização. O NetBeans cria para cada projeto uma pasta particular.

Desmarque o campo **Criar classe principal**. Clique em finalizar para concluir o procedimento de criação do projeto. Serão apresentadas as informações do projeto criado na aba Projeto.

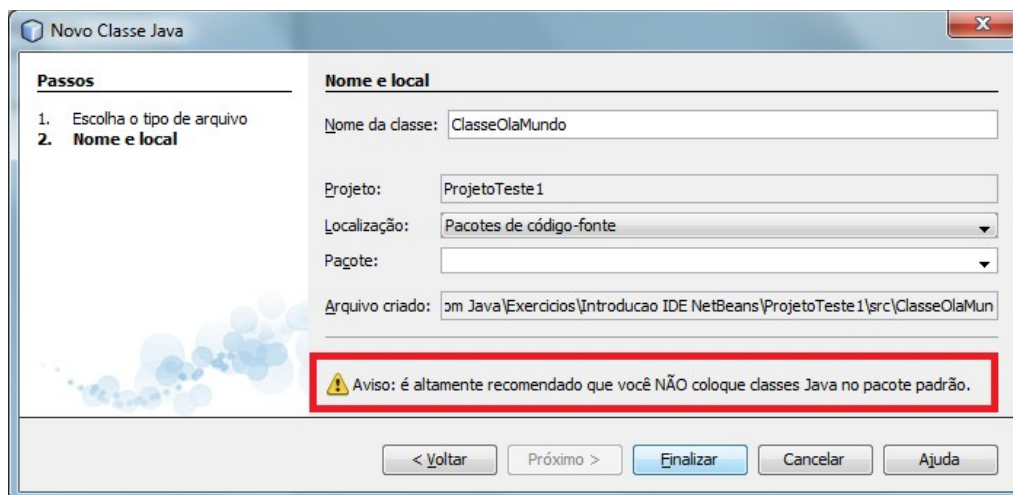


Roteiro de Uso da Ferramenta Netbeans

Criar uma classe para o projeto. Clique com o botão direito do mouse sobre o nome do projeto e escolha as opções conforme a imagem a seguir.

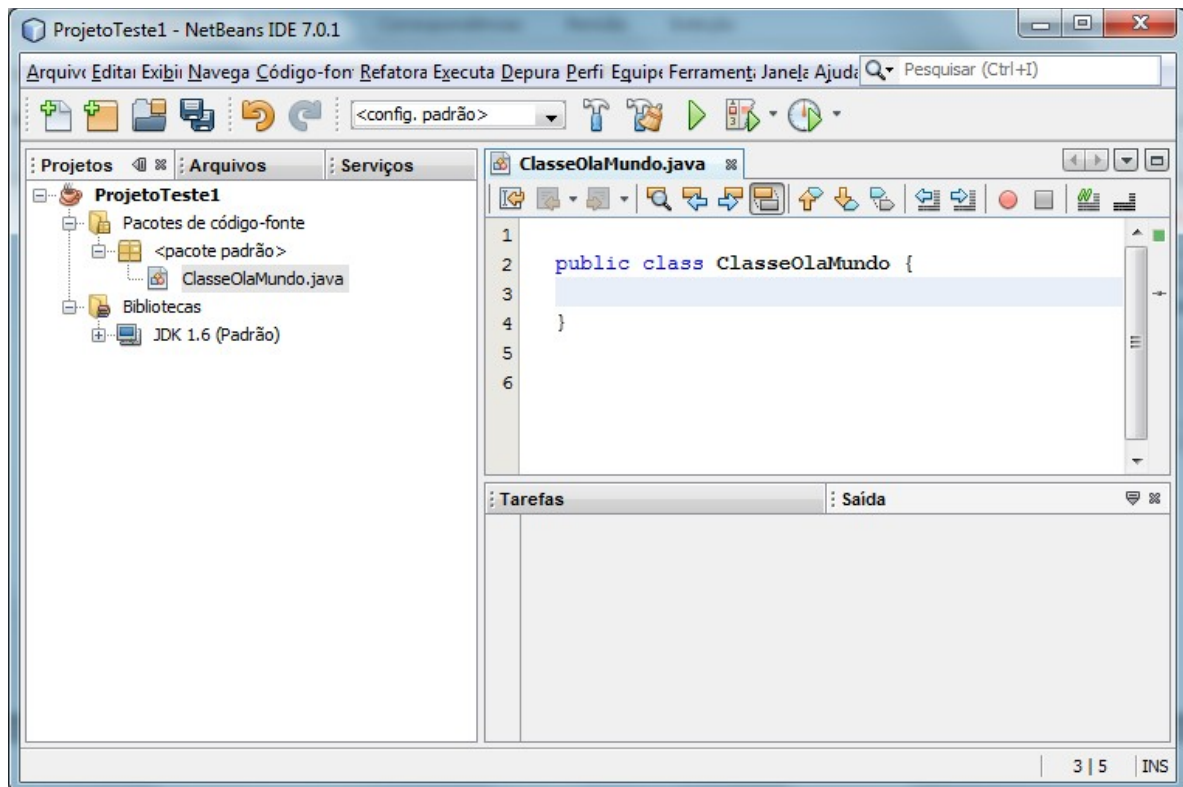


Será apresentada a tela a seguir:



Clique em finalizar. Veja a classe criada no projeto.

Roteiro de Uso da Ferramenta Netbeans

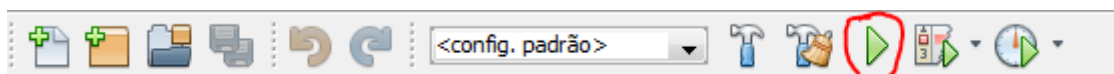


Preencha o corpo da classe com o seguinte conteúdo:

```
public class ClasseOlaMundo {  
    public static void main (String args[]) {  
        System.out.println("Olá, esse é um exemplo de classe.");  
    }  
}
```

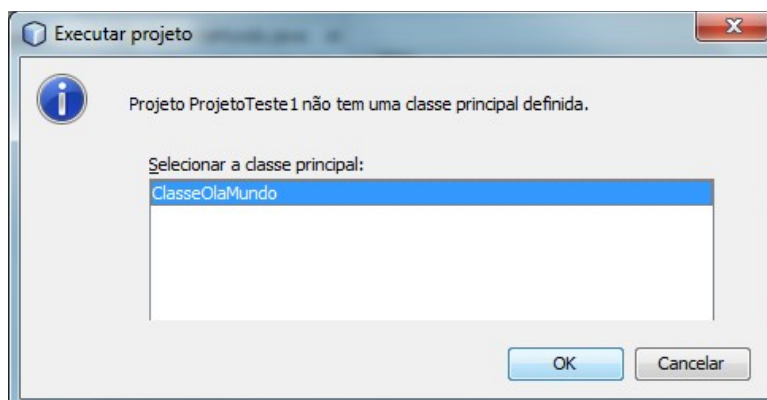
Para compilarmos o projeto (gerar os bytes code do Java) e executarmos o projeto podemos usar vários caminhos:

- Acesse o menu Executar e escolha a opção Executar projeto principal.
- Tecla de atalho F6 para Executar o projeto principal.
- Por meio do botão de atalho na barra de botões, passe o mouse sobre os botões para que seja exibida a dica de tarefa de cada botão.



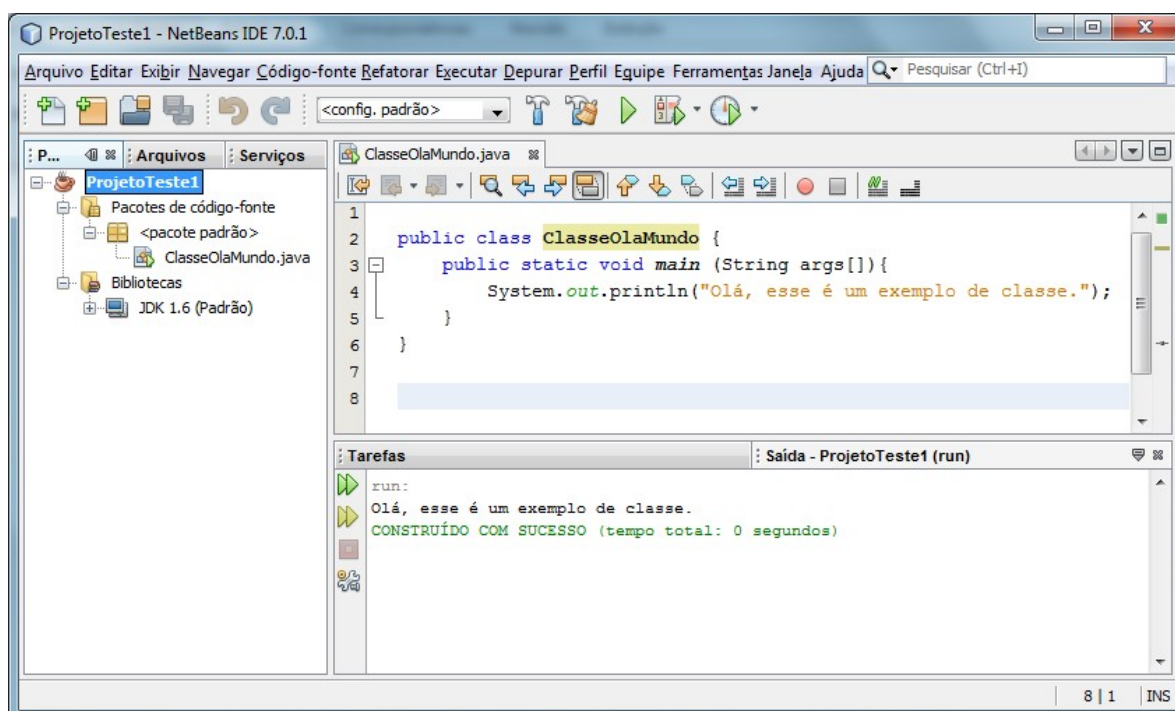
Roteiro de Uso da Ferramenta Netbeans

Na primeira vez que o projeto é executado, será solicitado que se defina qual é a classe principal do projeto. Todas as classes que possuem o método main, são candidatas a serem classe principal do projeto.



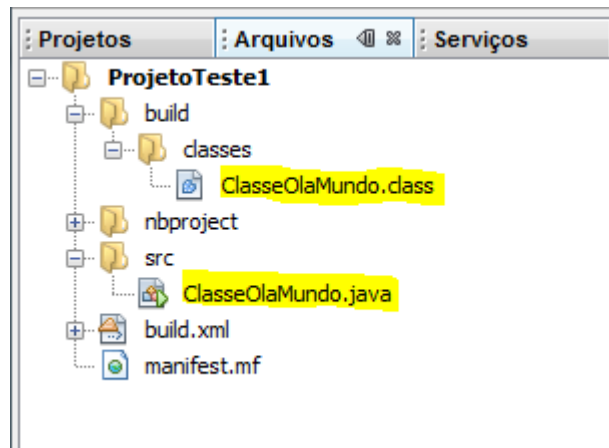
Obs: Existe a opção de menu Construir Projeto Principal que executa a compilação dos arquivos Java que ainda não foram compilados. Quando o projeto já foi executado mais de uma vez é mais seguro usar a opção Limpar e Construir, pois garante que todas as alterações em códigos-fonte serão consideradas na próxima execução.

Com a execução do projeto será exibido o resultado na área de saída da IDE, veja figura a seguir:

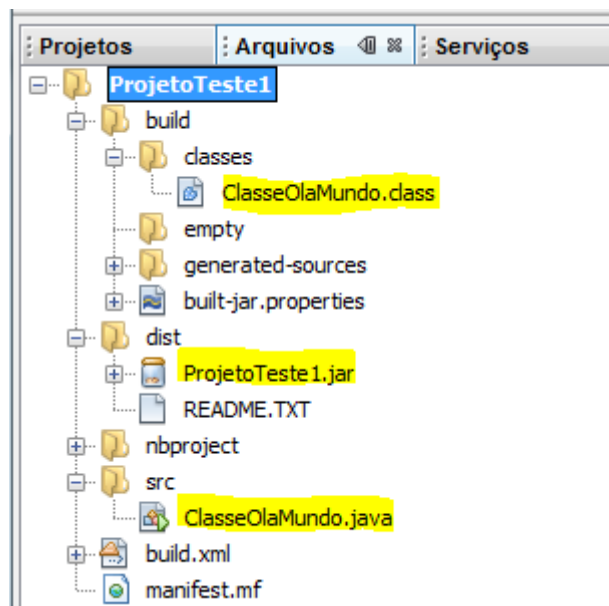


Roteiro de Uso da Ferramenta Netbeans

Após a execução é possível observar o arquivo .class gerado pela compilação. Use a guia Arquivos para visualizar.



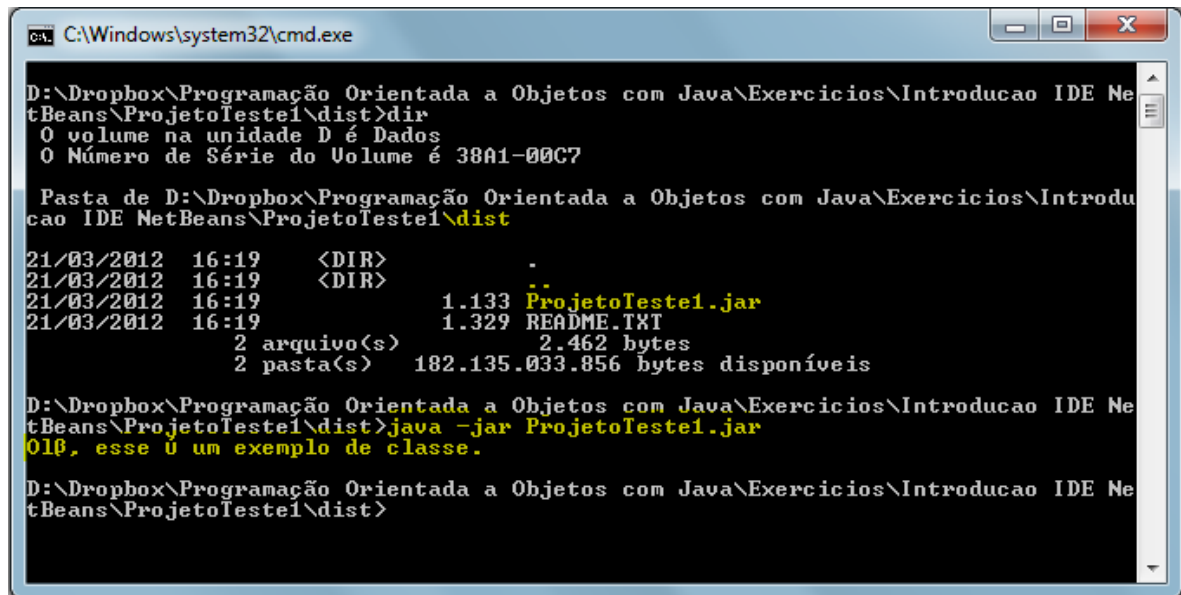
Quando usamos a opção Limpar e Construir Projeto Principal, cada classe compilada pode gerar um ou mais arquivos .class. Em seguida é gerado um arquivo .jar na pasta dist do projeto, que é a compactação de todos os arquivos necessários a execução do projeto fora da IDE. Veja imagem a seguir, que mostra a guia Arquivos depois da construção do ProjetoTeste1.



Duas pastas são criadas na construção, a pasta build que contém as classes compiladas, os arquivos .class e a pasta dist, que contém o arquivo jar, que é o arquivo de distribuição do projeto.

Roteiro de Uso da Ferramenta Netbeans

Executando o projeto fora da IDE por meio do aplicativo Java.exe (chama a JVM para executar o projeto).



```
C:\Windows\system32\cmd.exe

D:\Dropbox\Programação Orientada a Objetos com Java\Exercicios\Introducao IDE Ne
tBeans\ProjetoTeste1\dist>dir
O volume na unidade D é Dados
O Número de Série do Volume é 38A1-00C7

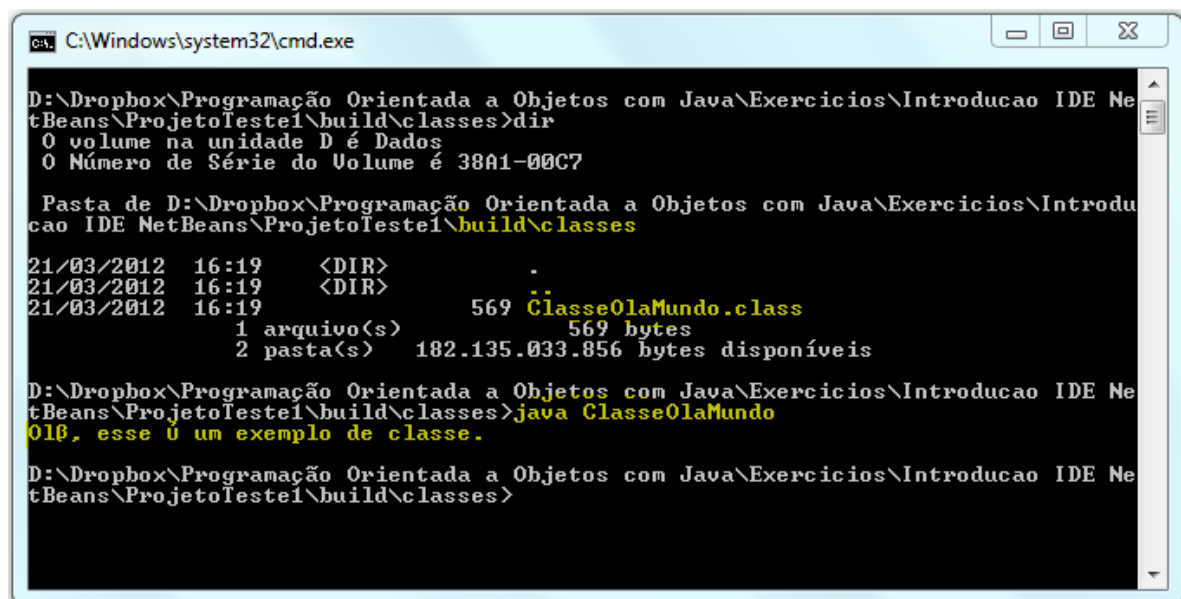
Pasta de D:\Dropbox\Programação Orientada a Objetos com Java\Exercicios\Introdu
cao IDE NetBeans\ProjetoTeste1\dist

21/03/2012  16:19    <DIR>          .
21/03/2012  16:19    <DIR>          ..
21/03/2012  16:19                1.133  ProjetoTeste1.jar
21/03/2012  16:19                1.329  README.TXT
                2 arquivo(s)          2.462 bytes
                2 pasta(s) 182.135.033.856 bytes disponíveis

D:\Dropbox\Programação Orientada a Objetos com Java\Exercicios\Introducao IDE Ne
tBeans\ProjetoTeste1\dist>java -jar ProjetoTeste1.jar
Olá, esse é um exemplo de classe.

D:\Dropbox\Programação Orientada a Objetos com Java\Exercicios\Introducao IDE Ne
tBeans\ProjetoTeste1\dist>
```

Executando apenas a classe ClasseOlaMundo.class compilada e armazenada na pasta build\classes.



```
C:\Windows\system32\cmd.exe

D:\Dropbox\Programação Orientada a Objetos com Java\Exercicios\Introducao IDE Ne
tBeans\ProjetoTeste1\build\classes>dir
O volume na unidade D é Dados
O Número de Série do Volume é 38A1-00C7

Pasta de D:\Dropbox\Programação Orientada a Objetos com Java\Exercicios\Introdu
cao IDE NetBeans\ProjetoTeste1\build\classes

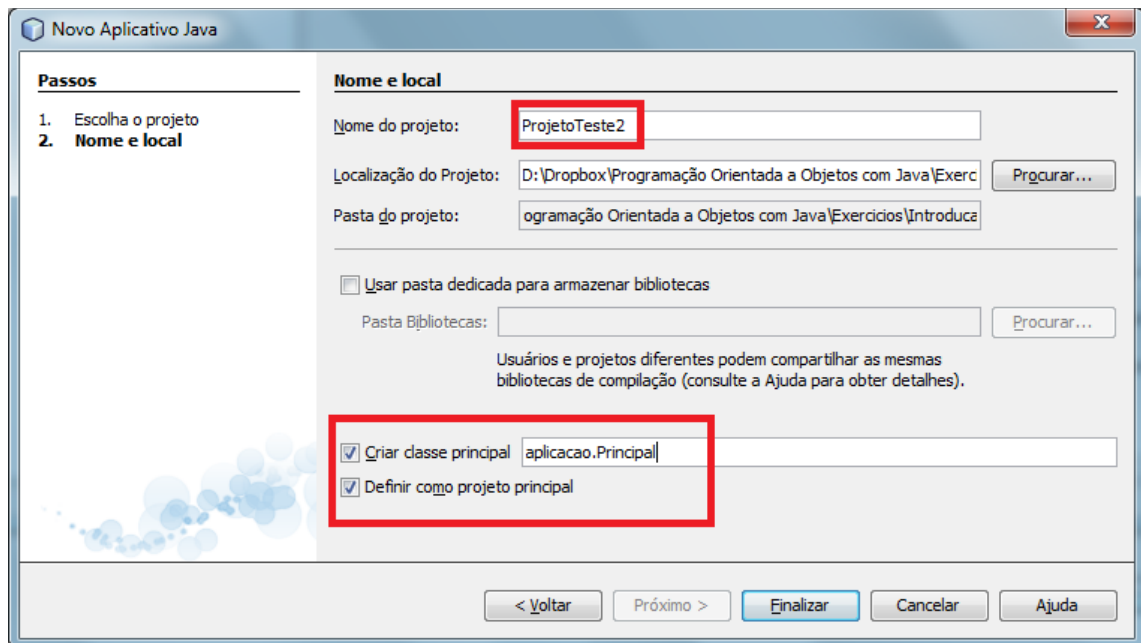
21/03/2012  16:19    <DIR>          .
21/03/2012  16:19    <DIR>          ..
21/03/2012  16:19                569  ClasseOlaMundo.class
                1 arquivo(s)          569 bytes
                2 pasta(s) 182.135.033.856 bytes disponíveis

D:\Dropbox\Programação Orientada a Objetos com Java\Exercicios\Introducao IDE Ne
tBeans\ProjetoTeste1\build\classes>java ClasseOlaMundo
Olá, esse é um exemplo de classe.

D:\Dropbox\Programação Orientada a Objetos com Java\Exercicios\Introducao IDE Ne
tBeans\ProjetoTeste1\build\classes>
```

Projeto 02

Crie um novo projeto na IDE, veja imagem a seguir:

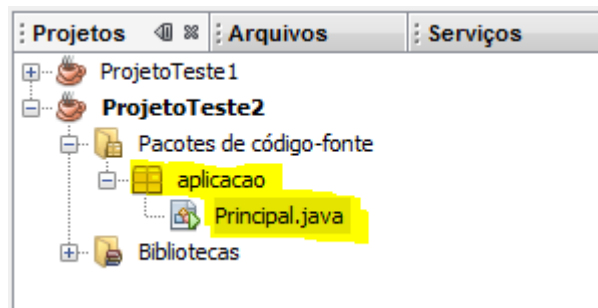


A opção Criar classe principal irá criar uma classe automaticamente e defini-la como classe principal do projeto.

A opção Definir como projeto principal é uma opção, pois podem existir vários projetos abertos e deseja-se que o que está sendo criado torne-se o projeto selecionado na IDE.

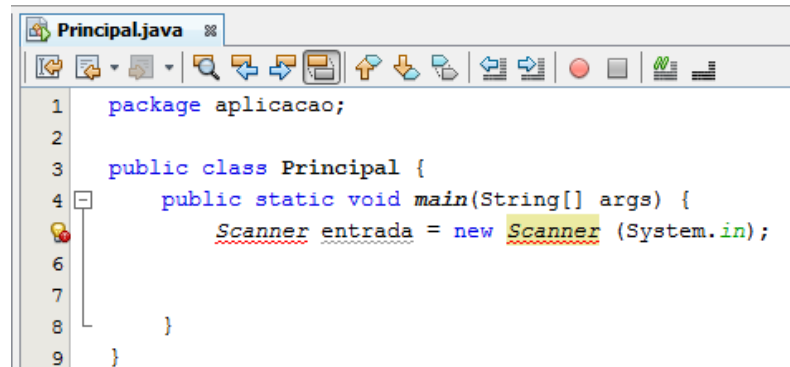
Clique em Finalizar. Veja a estrutura do projeto criado. A classe Principal.java foi criada no pacote aplicação.

Observação: nomes de pacotes devem ser sempre escrito em minúsculo.

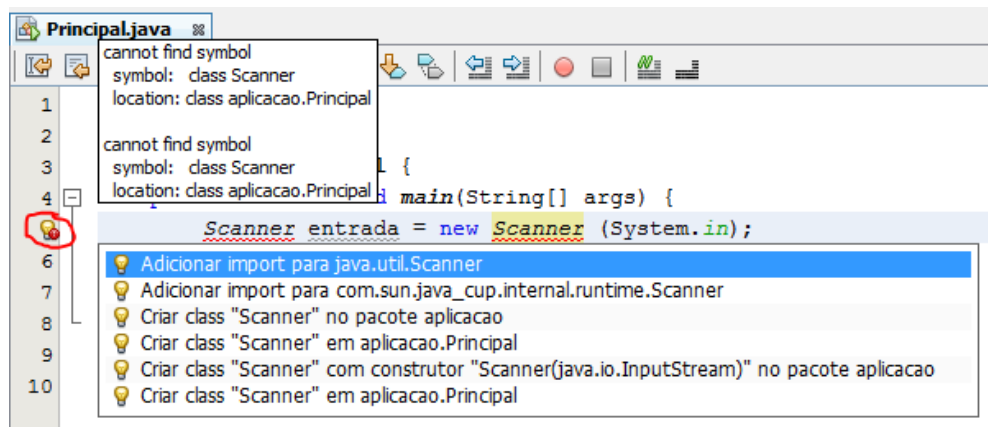


Roteiro de Uso da Ferramenta Netbeans

Vamos editar a classe Principal. Digite a linha de comando conforme figura a seguir:

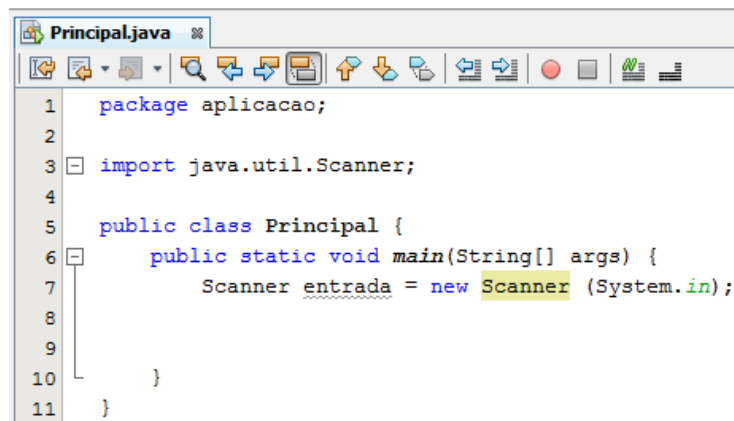


Repare que a IDE sublinha a palavra Scanner em vermelho, isso significa que não está reconhecendo esse identificador ou é necessário importar essa classe. Nesse caso é necessário importar a classe Scanner, pois ela não pertence ao pacote padrão java.lang do Java. É possível importar a classe de maneira manual, escrevendo a diretiva *import* no cabeçalho da classe, ou usar a IDE para fazer isso.



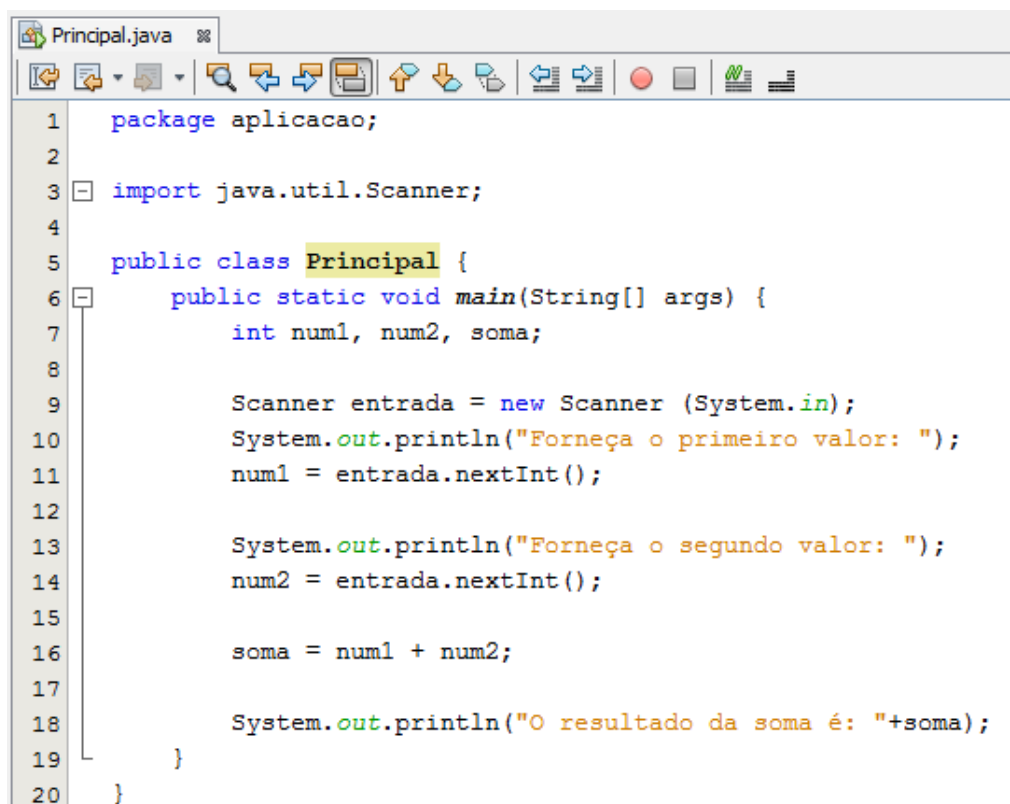
Depois da inclusão da diretiva *import*, a classe é reconhecida.

Roteiro de Uso da Ferramenta Netbeans



```
1 package aplicacao;
2
3 import java.util.Scanner;
4
5 public class Principal {
6     public static void main(String[] args) {
7         Scanner entrada = new Scanner (System.in);
8
9
10    }
11 }
```

Digite o restante do método main(), conforme figura a seguir:



```
1 package aplicacao;
2
3 import java.util.Scanner;
4
5 public class Principal {
6     public static void main(String[] args) {
7         int num1, num2, soma;
8
9         Scanner entrada = new Scanner (System.in);
10        System.out.println("Forneça o primeiro valor: ");
11        num1 = entrada.nextInt();
12
13        System.out.println("Forneça o segundo valor: ");
14        num2 = entrada.nextInt();
15
16        soma = num1 + num2;
17
18        System.out.println("O resultado da soma é: "+soma);
19    }
20 }
```

Roteiro de Uso da Ferramenta Netbeans

Nova estrutura para a classe principal.

```
package aplicacao;
import java.util.Scanner;
public class Principal {

    int num1, num2, soma;

    void lerDados() {
        Scanner entrada = new Scanner (System.in);
        System.out.print("Forneça o primeiro valor: ");
        num1 = entrada.nextInt();

        System.out.print("Forneça o segundo valor: ");
        num2 = entrada.nextInt();
    }

    int somar(){
        return num1 + num2;
    }

    void mostrarResultado() {
        System.out.println("O resultado da soma é: "+somar());
    }

    public static void main(String[] args) {
        Principal principal = new Principal();
        principal.lerDados();
        principal.mostrarResultado();
    }
}
```

1 Como criar um novo projeto Java no NetBeans

Para criarmos um projeto novo no NetBeans podemos usar a opção de menu Arquivo / Novo Projeto, em seguida será apresentada a tela a seguir, veja imagem:

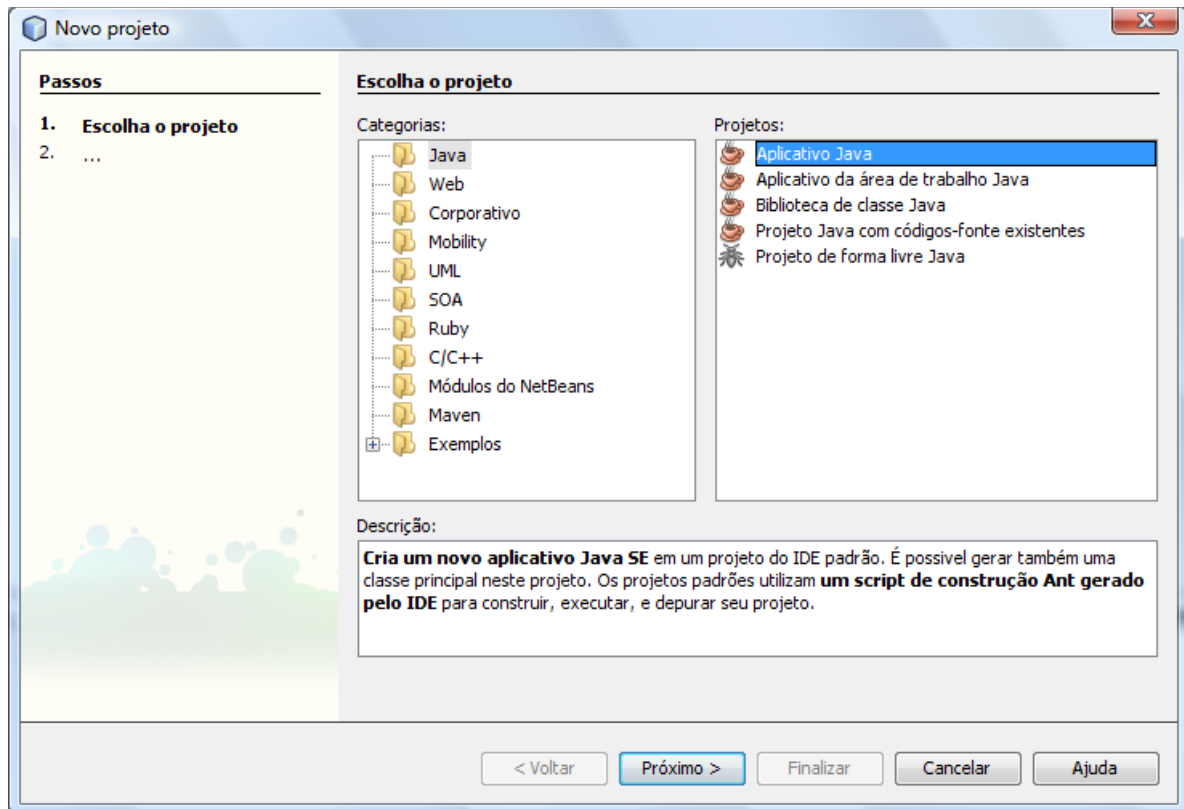


Figura 1. Novo Projeto Java

O segundo passo é escolher a categoria do projeto a ser criando, no nosso caso usaremos a categoria de projetos Java e o tipo de projeto Aplicativo Java. Clique no botão Próximo e será apresentada a tela a seguir:

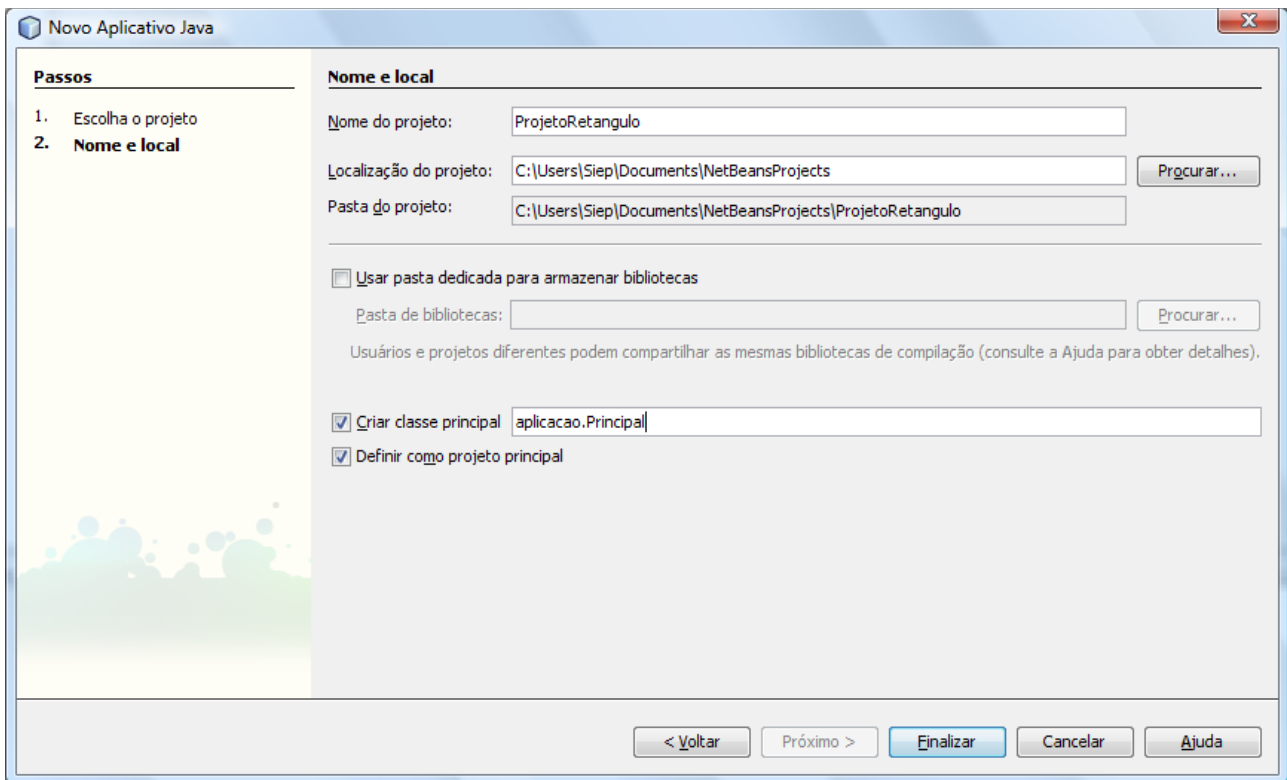
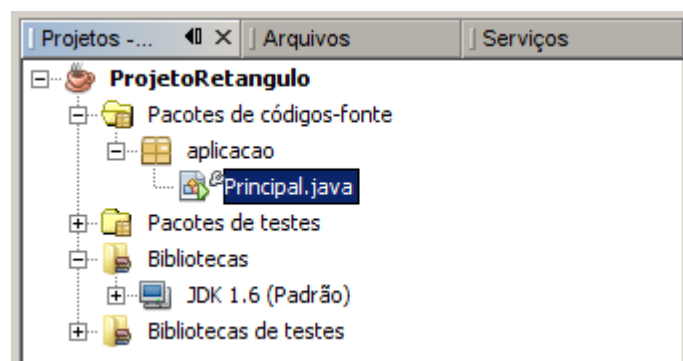


Figura 2. Definição de Características do Novo Projeto

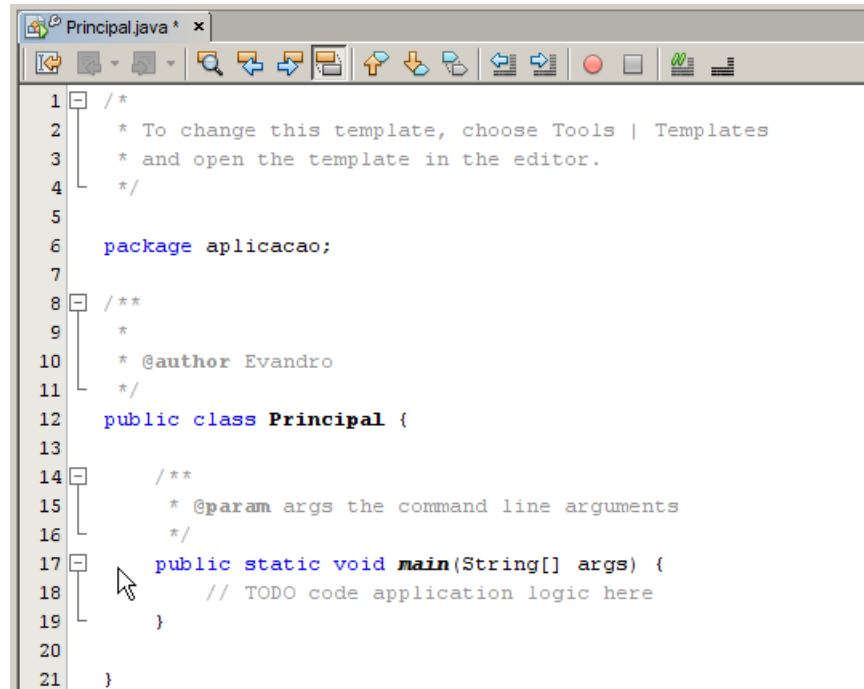
Na tela anterior devemos informar o nome do projeto a ser criado (ProjetoRetangulo), a localização do projeto, ou seja, onde o projeto será armazenado. O campo pasta do projeto será gerado automaticamente em função do nome do projeto e da localização. O NetBeans cria para cada projeto uma pasta particular.

Iremos alterar o campo Criar classe principal para aplicacao.Principal, onde aplicacao é nome do pacote que uma classe denominada Principal irá ser criada.



Podemos observar o ProjetoRetangulo dividido em quatro partes: Pacotes de código-fonte, Pacotes de testes, Bibliotecas e Bibliotecas de testes. Para nós nesse momento nos interessam duas partes: os Pacotes de código-fonte (as nossas classes) e as Bibliotecas que embora não precisamos alterar nada do que o NetBeans faz, mas é importante a consciência de que o projeto usa a biblioteca de classes do Kit de Desenvolvimento Java versão 1.6.

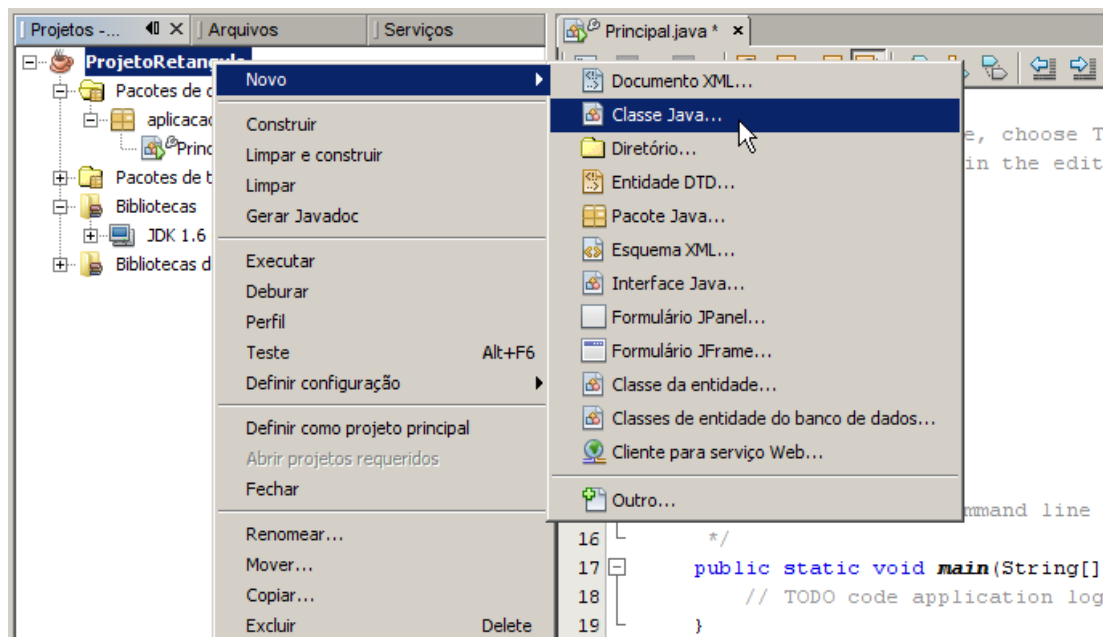
Observe que a classe que configuramos durante os procedimentos de criação do projeto foi criada e está armazenada nos Pacotes de código-fonte no pacote aplicacao. O código-fonte da classe pode ser visualizado na área de edição, que também é organizada em guias, cada uma edita um arquivo, veja imagem a seguir:



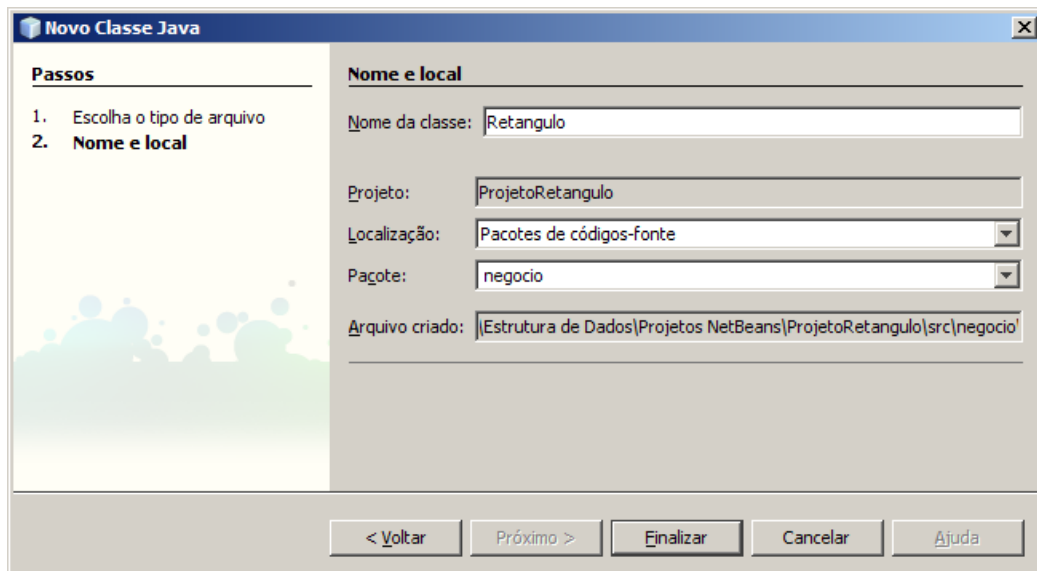
Trata-se de uma classe que possui apenas um método (método main) com o corpo vazio, isso significa que a lógica de execução deve ser definida por nós.

Antes de editarmos a classe Principal, vamos criar outra classe, uma classe de negócio que representa um conceito do mundo real.

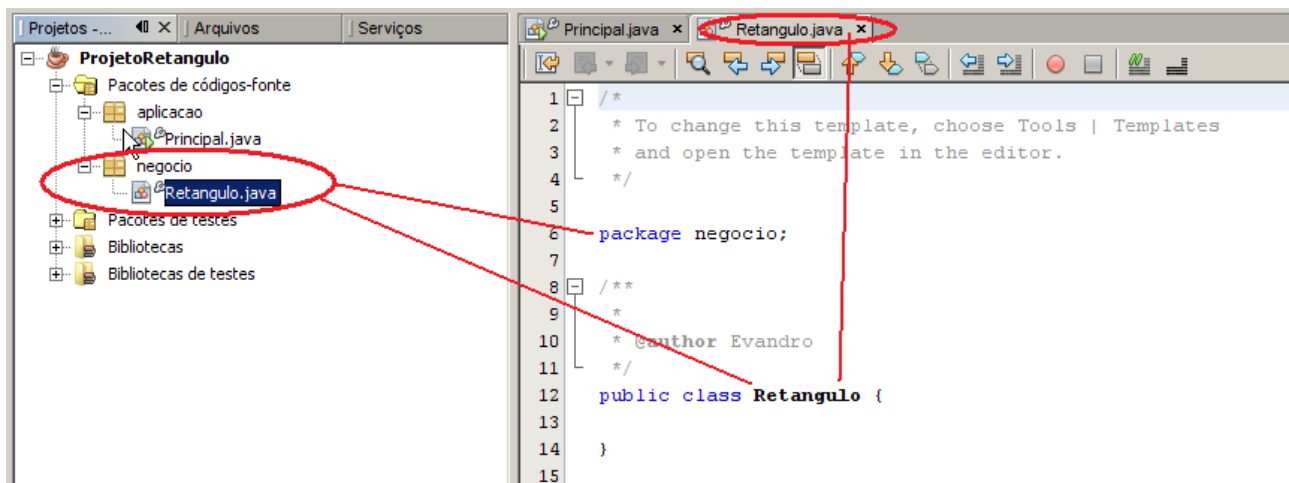
Nesse nosso exemplo, será a classe Retângulo. Para isso clique com o botão direito do mouse sobre o nome do projeto e escolha a opção Novo, em seguida a opção Classe Java, veja imagem a seguir.



Será apresentada a tela de configuração da nova classe, veja imagem a seguir:

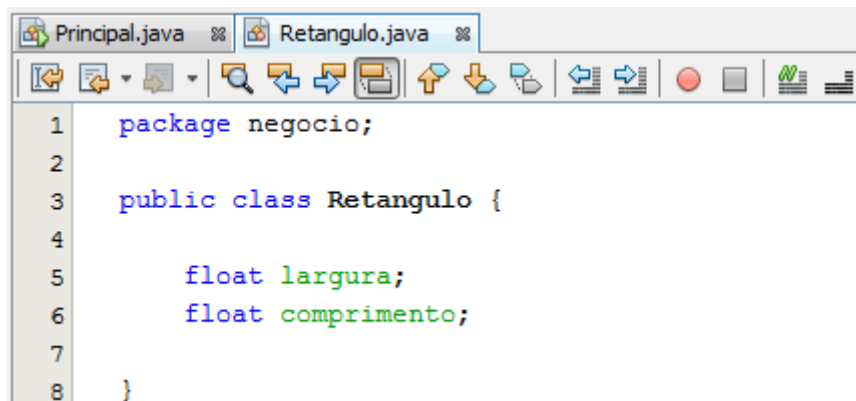


Nessa tela deve ser informado o nome da nova classe (Retangulo) e o nome do pacote que ela deve ser armazenada (negocio). Clique em Finalizar e a classe será criada. Veja na próxima imagem os novos elementos criados e suas relações:

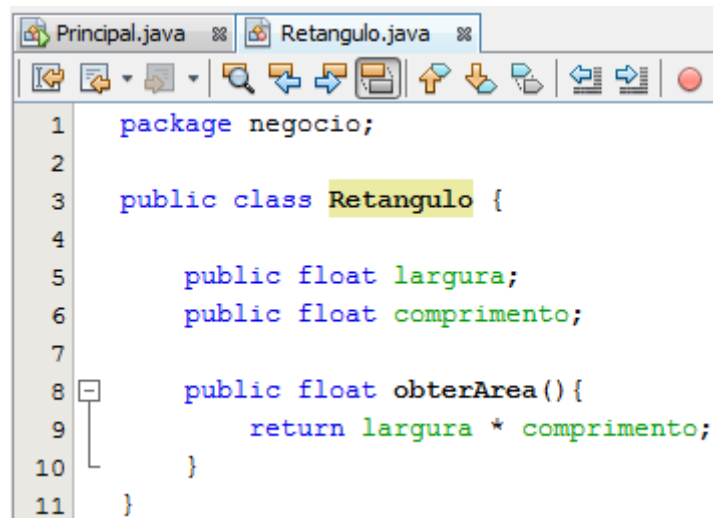


Foi criado um novo pacote (negocio) onde a classe Retangulo foi armazenada, repare na diretiva package no código-fonte da classe, indicando qual pacote ela pertence. Agora nosso projeto possui duas classes, uma no pacote aplicacao (Principal) e outra no pacote negocio (Retangulo). Agora vamos inserir a lógica dessa aplicação, começaremos pela classe Retangulo.

Começaremos pela definição dos atributos da classe Retangulo, serão inseridos dois atributos: altura e comprimento, ambos do tipo float (real em Java). Veja imagem a seguir

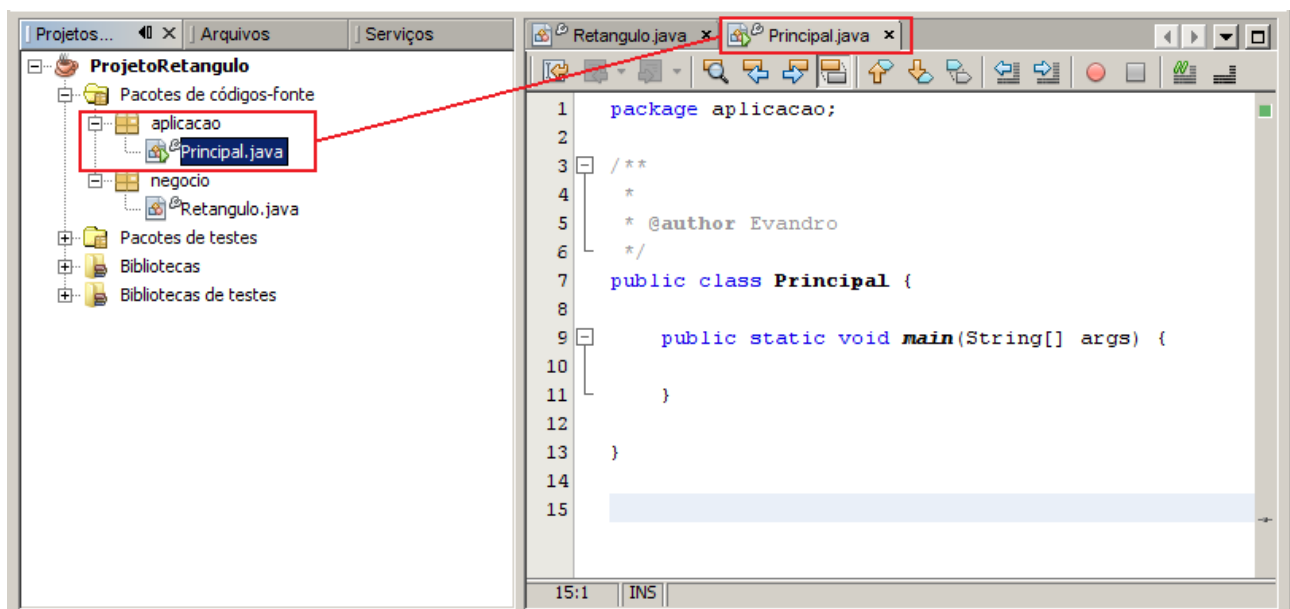


Vamos agora inserir um método de processamento, o método obterArea, que irá calcular a área do retângulo através da largura e comprimento. Esse método deverá ser escrito manualmente. Veja a imagem a seguir:



```
1 package negocio;
2
3 public class Retangulo {
4
5     public float largura;
6     public float comprimento;
7
8     public float obterArea() {
9         return largura * comprimento;
10    }
11 }
```

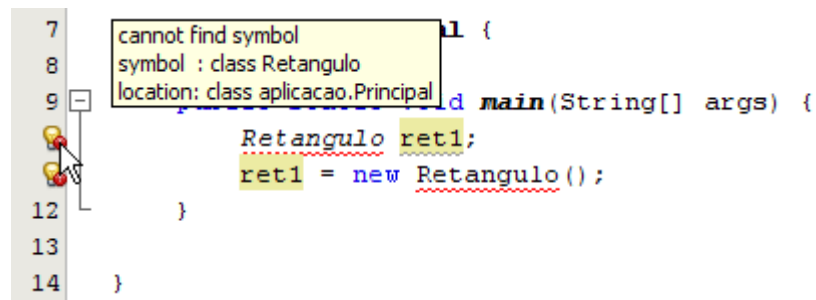
Agora já temos a nossa classe de negócio (Retangulo), cada objeto de Retangulo será capaz de representar a largura e o comprimento de um retângulo e calcular a área através desses valores. Vamos agora escrever o código da classe Principal. Para selecionar a classe Principal, vá a guia de Projetos abra o pacote aplicação (caso não esteja), e dê um duplo clique no nome da classe Principal, isso abrirá e selecionará a guia do código dessa classe, veja figura a seguir:



Digite o código do método main da classe Principal, conforme trecho de código a seguir:

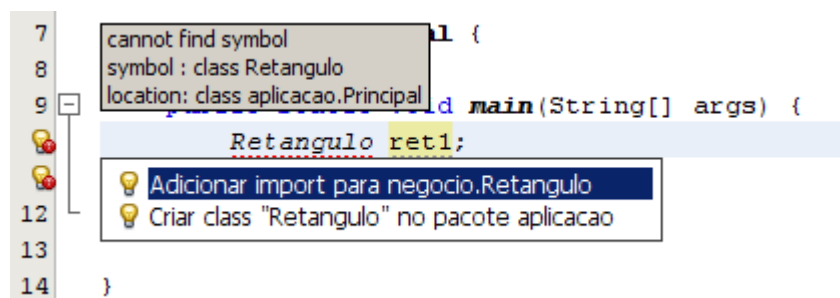
```
7 public class Principal {  
8  
9     public static void main(String[] args) {  
10         Retangulo ret1;  
11         ret1 = new Retangulo();  
12     }  
13  
14 }
```

Repare que após a digitação dessas duas linhas o NetBeans marcou com vermelho a classe Retangulo e seu construtor. Essa é uma excelente característica do NetBeans (e também de boa parte das IDE's Java). Sempre que o NetBeans não reconhece um identificador em seu código ele irá marcá-lo, repare também que do lado esquerdo onde ficariam os números de linhas 10 e 11 foram substituídos por uma lâmpada e um símbolo de erro, isso significa que a IDE encontrou um erro, mas tem uma ou mais sugestões de acerto do erro, coloque o cursor do mouse sobre uma das lâmpadas (sem clicar) e uma mensagem será apresentada, veja imagem a seguir:



The screenshot shows the same code as before, but with a tooltip displayed over the error icon on line 11. The tooltip text is: "cannot find symbol", "symbol : class Retangulo", and "location: class aplicacao.Principal".

A mensagem diz que não foi possível encontrar o símbolo classe Retangulo. Você deve estar pensando, mas acabamos de construir a classe Retangulo, qual será o motivo do erro? As classes Principal e Retangulo foram criadas em pacotes diferentes, Principal está no pacote aplicacao e Retangulo está no pacote negocio. Deve-se importar a classe Retangulo. Veja imagem a seguir:



The screenshot shows the same code with the error tooltip. A second tooltip is now visible below the first one, containing two suggestions: "Adicionar import para negocio.Retangulo" and "Criar class 'Retangulo' no pacote aplicacao".

Veja o resultado desse procedimento no código da classe Aplicacao.

```
1 package aplicacao;
2
3 import negocio.Retangulo;
4
5 /**
6  *
7  * @author Evandro
8  */
9 public class Principal {
10
11     public static void main(String[] args) {
12         Retangulo ret1;
13         ret1 = new Retangulo();
14     }
15
16 }
```

Observe que os erros desapareceram, pois o *import* resolveu os dois erros. Vamos agora continuar inserindo código no corpo do método main. Veja código a seguir:

```
package aplicacao;

import javax.swing.JOptionPane;
import negocio.Retangulo;

public class Principal {

    public static void main(String[] args) {

        Retangulo ret1;
        ret1 = new Retangulo();

        float larg, comp;
        String sLarg, sComp;

        sLarg = JOptionPane.showInputDialog("Forneça a largura do retângulo");
        sComp = JOptionPane.showInputDialog("Forneça o comprimento do retângulo");
```

O mesmo procedimento terá que ser feito para a classe *JOptionPane* que não foi reconhecida. Para resolver o problema repita o procedimento realizado para a classe *Retangulo*. Essa classe está sendo usada para realizar entrada de dados, é equivalente ao comando *leia()* da disciplina de algoritmo. Como foi dito no início desse roteiro, algumas coisas podem não ficar compreendidas em relação à sintaxe da linguagem Java, porém esses detalhes serão fornecidos na próxima unidade. Veja a seguir o código completo da classe *Principal*.

```
package aplicacao;

import javax.swing.JOptionPane;
import negocio.Retangulo;

public class Principal {

    public static void main(String[] args) {

        Retangulo ret1;
        ret1 = new Retangulo();

        float larg, comp;
        String sLarg, sComp;

        sLarg = JOptionPane.showInputDialog("Forneça a largura do retângulo");
        sComp = JOptionPane.showInputDialog("Forneça o comprimento do retângulo");

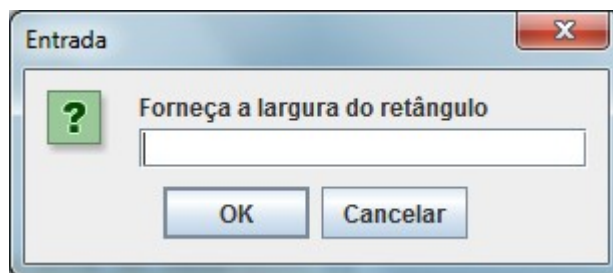
        larg = Float.parseFloat(sLarg);
        comp = Float.parseFloat(sComp);

        ret1.largura = larg;
        ret1.comprimento = comp;

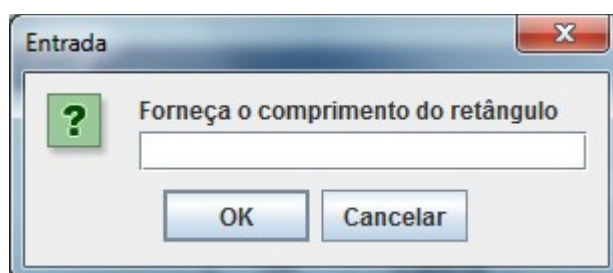
        JOptionPane.showMessageDialog(null, "A área do retângulo = "+ret1.obterArea());

    }
}
```

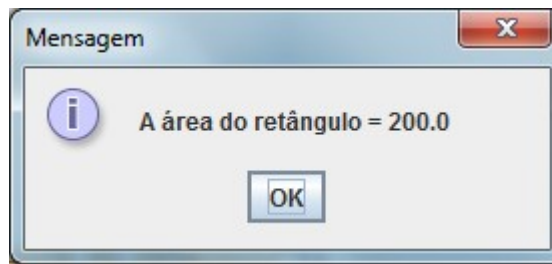
Quando executarmos esse nosso projeto exemplo, será exibida a tela que irá pedir para digitar o valor da largura do retângulo, veja imagem a seguir:



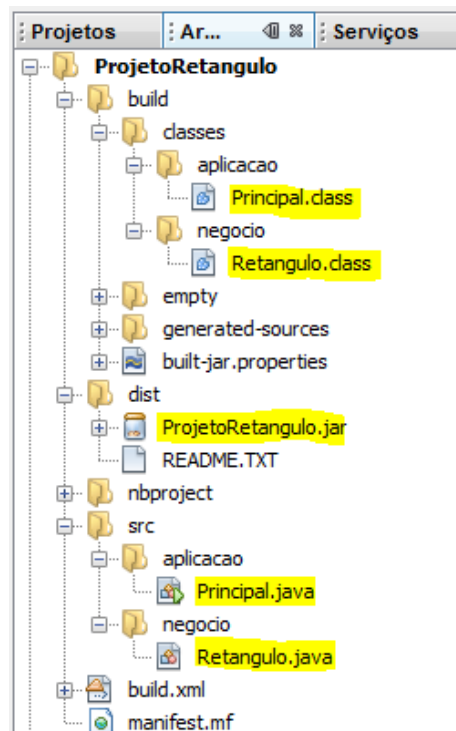
Forneça um valor e tecle Enter ou clique no botão OK para confirmar a entrada do valor. Em seguida será apresentada a tela que irá pedir para digitar o valor do comprimento do retângulo, veja imagem a seguir:



Digite o valor para o comprimento e novamente tecele Enter ou clique no botão OK para confirma o valor fornecido. Por último o programa irá mostrar uma tela contendo o resultado do cálculo da área do retângulo, veja imagem a seguir:



Quando usamos a opção Construir Projeto Principal (menu Executar), cada classe é compilada e um ou mais arquivos .class será gerado para cada arquivo .java. Em seguida é gerado um arquivo .jar na pasta dist do projeto, que é a compactação de todos os arquivos necessário a execução projeto fora da IDE. Veja imagem a seguir, que mostra a guia Arquivos depois da construção do ProjetoRetangulo.



Duas pastas são criadas na construção, a pasta build que contém as classes compiladas, os arquivos .class e a pasta dist, que contém o arquivo jar, que é o arquivo de distribuição do projeto. Você pode ver essa estrutura através de uma ferramenta de exploração de conteúdo de disco como por exemplo, o Windows Explorer, e ao explorar a pasta dist, dê um duplo clique no arquivo ProjetoRetangulo.jar, notará que será executado a aplicação construída nos passos anteriores. Você pode também copiar esse arquivo jar para outra máquina que possui a jvm do Java e testá-lo nessa máquina.

1 Como criar um novo projeto Java no NetBeans

Para criarmos um projeto novo no NetBeans podemos usar a opção de menu Arquivo / Novo Projeto. Escolha a categoria/projeto: Java / Aplicação Java. Clique no botão Próximo e será apresentada a tela a seguir:

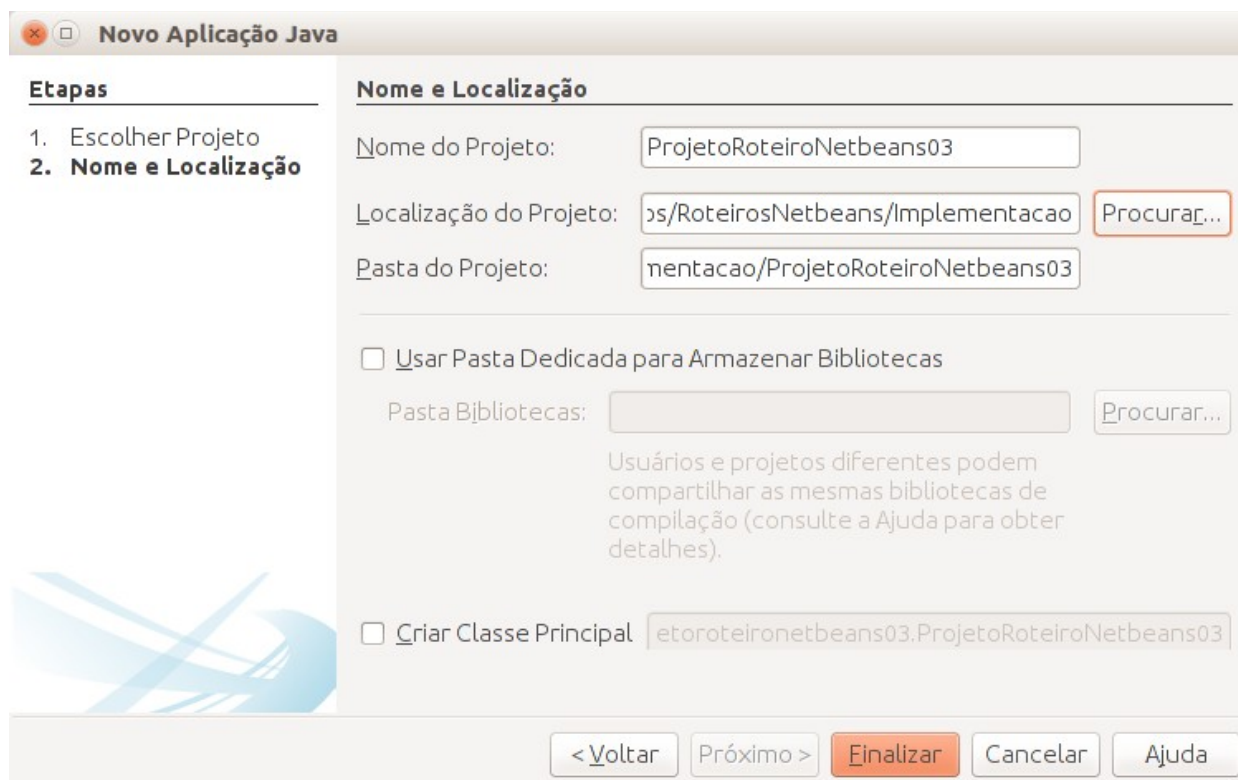
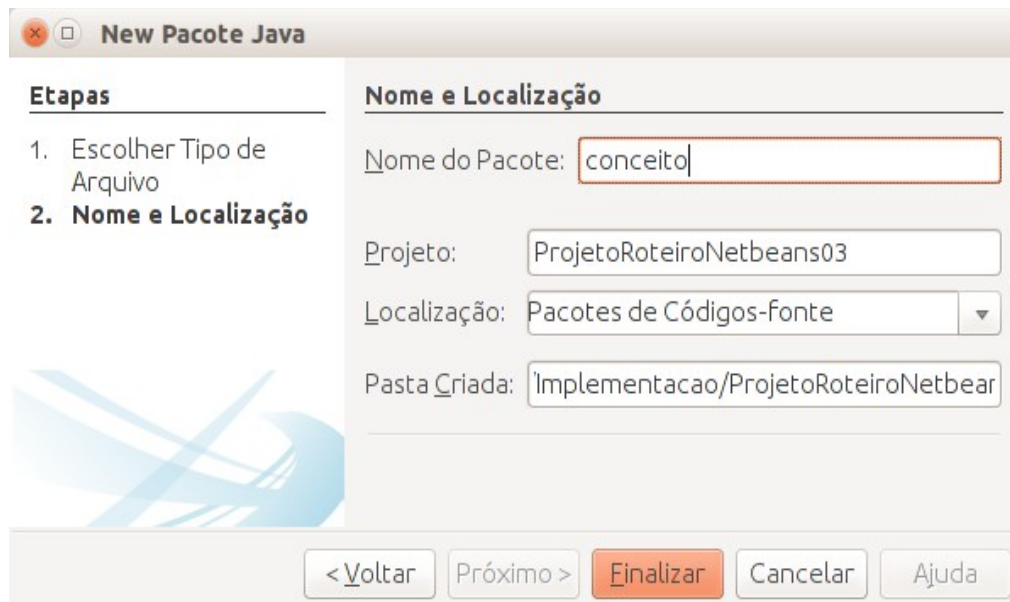


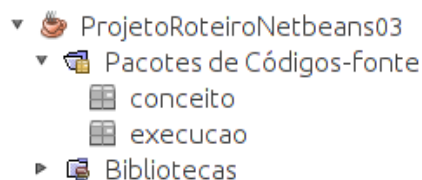
Figura 1. Definição de Características do Novo Projeto

Na tela anterior devemos informar o nome do projeto a ser criado (ProjetoRoteiroNetbeans03), a localização do projeto, ou seja, onde o projeto será armazenado. O campo pasta do projeto será gerado automaticamente em função do nome do projeto e da localização. O NetBeans cria para cada projeto uma pasta particular.

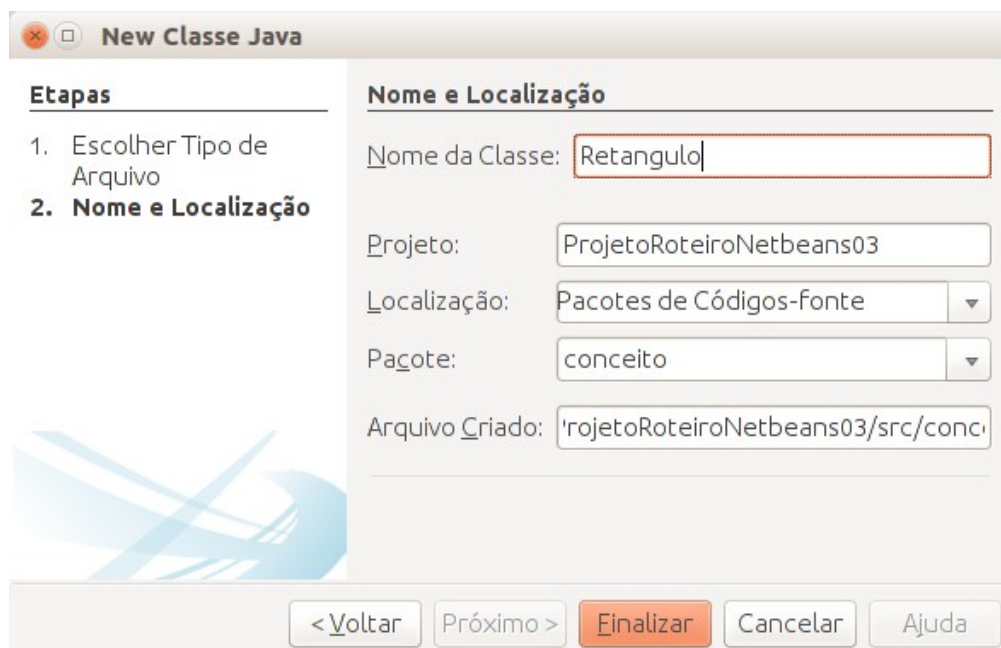
Serão criados dois pacotes para organizarem as classes da aplicação. Para isso, clique com o botão direito do mouse sobre o nome do projeto e escolha a opção Novo, em seguida a opção Pacote Java (package), veja imagem a seguir.



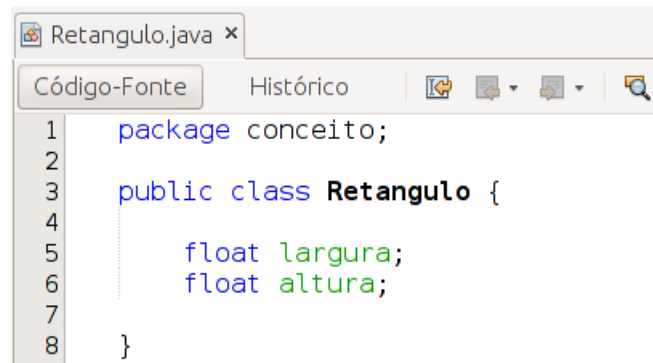
Defina o nome do pacote com o valor **conceito** e clique em Finalizar. Crie mais um pacote com o nome **execucao**. Após criar os dois pacotes, o projeto terá a seguinte estrutura:



Nesse nosso exemplo, será a classe Retângulo. Para isso clique com o botão direito do mouse sobre o pacote **conceito** e escolha a opção Novo, em seguida a opção Classe Java, veja imagem a seguir.

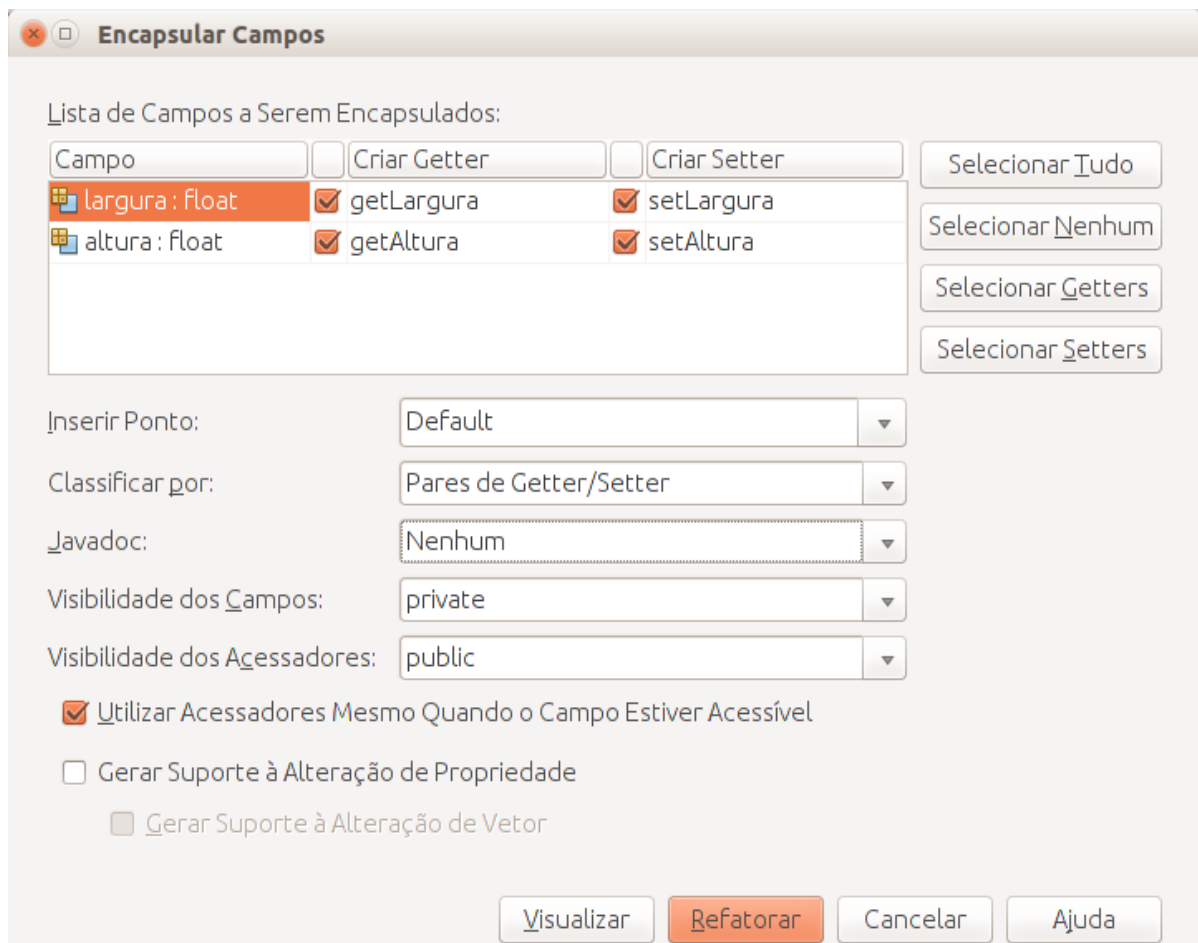


Nessa tela deve ser informado o nome da nova classe (Retangulo) e o nome do pacote será sugerido (negocio), pois foi escolhida a opção Novo a partir do pacote conceito. Clique em Finalizar e a classe será criada. Defina os atributos da classe Retangulo, serão inseridos dois atributos: altura e comprimento, ambos do tipo float (real em Java). Veja imagem a seguir



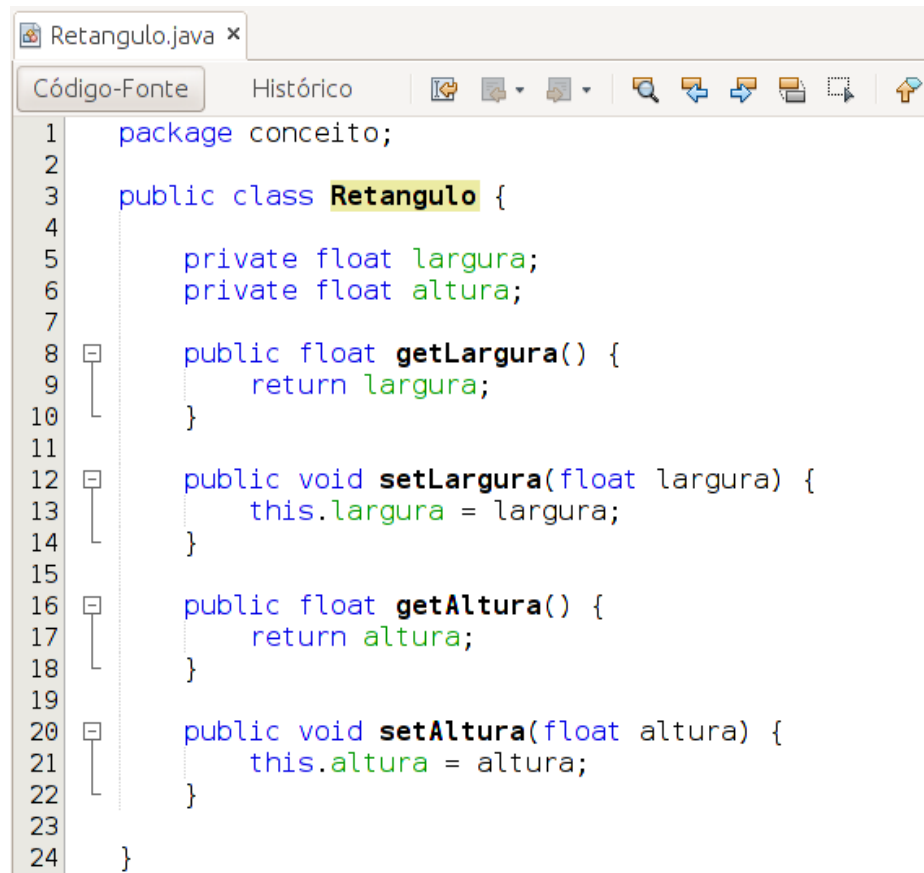
```
1 package conceito;
2
3 public class Retangulo {
4
5     float largura;
6     float altura;
7
8 }
```

Clique com o botão direito do mouse e escolha a opção Refatorar/Encapsular Campos. Será apresentada a tela a seguir:



Configure-a conforme a imagem e clique em Refatorar. Serão gerados os métodos de acesso Get`s e Set`s e os atributos serão definidos com visibilidade privada.

Veja as alterações na classe após a execução da refatoração. Essa operação pode ser executada pela opção Inserir Código / Getter e Setter.



```
1 package conceito;
2
3 public class Retangulo {
4     private float largura;
5     private float altura;
6
7     public float getLargura() {
8         return largura;
9     }
10
11     public void setLargura(float largura) {
12         this.largura = largura;
13     }
14
15     public float getAltura() {
16         return altura;
17     }
18
19     public void setAltura(float altura) {
20         this.altura = altura;
21     }
22 }
23
24 }
```

Vamos agora inserir um método de processamento, o método obterArea, que irá calcular a área do retângulo através da largura e comprimento. Esse método deverá ser escrito manualmente. Veja a imagem a seguir:

```
package conceito;

public class Retangulo {

    private float largura;
    private float altura;

    public float getLargura() {
        return largura;
    }

    public void setLargura(float largura) {
        this.largura = largura;
    }

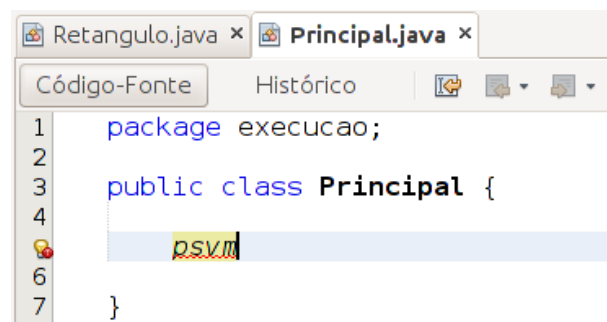
    public float getAltura() {
        return altura;
    }

    public void setAltura(float altura) {
        this.altura = altura;
    }

    public float obterArea(){
        return this.altura * this.largura;
    }

}
```

Agora já temos a nossa classe de negócio (Retangulo), cada objeto de Retangulo será capaz de representar a largura e o comprimento de um retângulo e calcular a área através desses valores. Vamos agora criar a classe Principal no pacote execucao.



Após ser criada a classe Principal, digite a abreviação **psvm** (public/static/void/main) conforme a figura e tecla **Tab**, será definido o método main na classe Principal, por meio da ferramenta auto completar (Código-Fonte/Auto Completar).

Digite o código do método main da classe Principal, conforme trecho de código a seguir:

```
package execucao;

import conceito.Retangulo;
import javax.swing.JOptionPane;

public class Principal {

    public static void main(String[] args) {

        Retangulo ret = new Retangulo();

        ret.setAltura(Float.parseFloat(JOptionPane.showInputDialog("Digite a altura")));
        ret.setLargura(Float.parseFloat(JOptionPane.showInputDialog("Digite a largura")));

        JOptionPane.showMessageDialog(null, "A area = " + ret.obterArea());

    }

}
```

Execute e teste a aplicação.

Outras opções de Auto Completar Código-Fonte:

if+Tab

ife+Tab

wh+Tab

do+Tab

sw+Tab

sout+Tab

serr+Tab