

Desenvolvimento Desktop

Aula 01 - Revis o de POO e Introdu o ao Netbeans

Apresentação

Nesta aula, revisaremos os conceitos de Orientação a Objetos (OO) vistos na disciplina de Programação Orientada a Objetos do Módulo Básico. Isso será muito importante, pois nesta disciplina você verá como esses conceitos podem ser utilizados na implementação de aplicações. Para esta aula, é importante que você estude bem os exemplos apresentados e os relacione com os conceitos mais importantes da OO, como classes, objetos, métodos, atributos e herança.

Além disso, você irá conhecer uma poderosa ferramenta de desenvolvimento chamada NetBeans IDE, que você utilizará a partir de agora para a criação de seus projetos em Java. O NetBeans IDE é uma ferramenta gráfica com recursos integrados para criar aplicações robustas e eficientes com uma maior facilidade. Seu editor de códigos é bem mais interativo do que o editor de texto que você vem utilizando até agora. Você sentirá a diferença à medida que for se acostumando com ele. Nesta aula aprenderemos como adquirir e instalar o Netbeans IDE, assim como aprenderemos a criar um projeto simples.



Vídeo 01 - Apresentação

Objetivos

Ao final desta aula, você será capaz de:

- Definir Orientação a Objetos.
- Conceituar Atributos e Métodos.
- Criar atributos e métodos em classes Java.
- Aplicar os conceitos de herança em Java.
- Entender o que é o NetBeans e por que utilizá-lo.
- Identificar quais os requisitos de hardware são necessários para a sua instalação.
- Instalar e executar o NetBeans.
- Criar seu primeiro projeto no Netbeans IDE.

Orientação a Objetos

A Orientação a Objetos (OO) é uma terminologia utilizada amplamente em programação que procura associar a lógica dos algoritmos com as estruturas de dados que esses utilizam. Para tanto, as operações e os dados de algoritmos são agrupados em entidades chamadas classes. Cada classe estabelece um conceito a ser implementado pelo algoritmo. Portanto, algoritmos implementados sob a ótica da OO definem classes de acordo com os conceitos que esses devem implementar; obviamente, quanto mais conceitos o algoritmo deve tratar, mais classes terão que ser definidas. Observe que, ao definir os conceitos de algoritmos, as classes acabam por definir a própria estrutura de dados deles. Pode-se, então, pensar em classes como estruturas de dados, com as funções associadas.

As variáveis de classes recebem o nome de **atributos** e as funções, por sua vez, recebem o nome de **métodos**. Por exemplo, um programa que precisar representar automóveis pode estabelecer como conceitos:

- Rodas
- Motor
- Volante
- Acelerador, etc

Cada um desses conceitos será implementado com uma classe em Java, da seguinte forma:

```
1 Arquivo Motor.java
2 public class Motor {
3     private double cilindrada;
4     private int potencia;
5     private int temperatura;
6
7     Motor(double _cilindrada) {
8         cilindrada = _cilindrada;
9         temperatura = 0;
10    }
11
12    public int getTemperatura() {
13        return temperatura;
14    }
15 }
```

É bom para a prática de programação iniciar os nomes de classes com letras maiúsculas, isso para diferenciar nomes de classes com nomes de instâncias (as instâncias, por convenção, são iniciadas sempre com letras minúsculas). Uma instância é a declaração de uma classe: instâncias são também chamadas de **objetos**, daí o nome de orientação a objetos a esse paradigma de programação.

Por exemplo, um objeto para motor pode ser declarado como:

```
1 Motor motor = new Motor(2.0); // declaração de um objeto do tipo Motor, passando o parâmetro 2
2
3 Motor motorP; // declaração de uma referência para um objeto do tipo Motor.
4
5 motorP = new Motor(3.6); // alocação de memória para o ponteiro e passagem de parâmetro para
```

Os acessos aos atributos e métodos se dão pelo operador ponto "."

```
1 motor.getTemperatura();
```

Observe que a classe **Motor** tem um método com o mesmo nome. Esse é particularmente chamado de método **construtor** da classe. Esse método serve para inicializar os atributos das instâncias da respectiva classe. O construtor é executado apenas uma vez, preenchendo os atributos com os valores passados por parâmetro para ele (é o caso dos valores 2.0 e 3.6 nos exemplos acima).

Outro ponto que deve ser observado é que existem as palavras **public** e **private** na declaração dos métodos e atributos. Você deve recordar da disciplina de programação OO que esses modificadores definem a maneira como são acessados

os métodos e atributos. Por exemplo, atributos e métodos privados (`private`) somente podem ser acessados por métodos da própria classe. Já os atributos e métodos públicos (`public`) podem ser acessados em qualquer lugar e a qualquer momento da execução do programa.

Atividade 01

1. Como se chamam as variáveis e as funções de classes?
2. O que você entende por instâncias e como podem ser também chamadas?
3. Que nome se dá ao método utilizado para inicializar atributos de uma instância?
4. O que você entende por variáveis públicas e privadas?

Construtores

A ideia principal por trás da OO é separar claramente os conceitos dos algoritmos em entidades **atômicas**, de modo a tornar o código mais organizado, legível e fácil de manter. Entende-se aqui por entidades **atômicas** aquelas que são autossuficientes, ou seja, podem existir por si só. Justamente por serem atômicas é que o acesso aos atributos internos das classes, se bem controlados, deixam o código mais seguro.

No exemplo anterior, a seguinte instrução, quando implementada em uma classe diferente da classe **Motor**, resultaria em erro:

```
1 motor.potencia = 10; // erro porque o atributo potencia é privado.
```

Mas, então, você poderia perguntar: como modificar e acessar o conteúdo de potência? A resposta mais apropriada, mantendo a atomicidade da classe, seria: por meio de métodos específicos para isso, como, por exemplo:

```
1 public int getPotencia() { // obtem potencia
2     return(potencia);
3 }
```

e

```
1 public void setPotencia(int _p) { // atribui potencia
2     potencia = _p;
3 }
```

Para modificar o valor do atributo **potencia**, faríamos:

```
1 motor.setPotencia(10); // Maneira correta de atribuir um valor a uma variável privada.
```

Dessa forma, somente **métodos da classe** podem acessar seus atributos, garantindo maior segurança na execução do algoritmo.

Normalmente, os atributos são declarados como privados e os métodos como públicos. Atente para o fato de que pelo menos um método da classe, necessariamente, precisa ser público; caso contrário, os objetos dessas classes não poderão interagir com o resto do programa. Assim como para atributos, métodos privados somente podem ser acessados por outros métodos da mesma classe.

Obviamente, como conceitos não existem sozinhos, as classes necessitam se relacionar com outras classes. Por exemplo, os conceitos apresentados anteriormente para a criação/definição de carros (rodas, motor, carenagem, volante e acelerador) necessitam se relacionar de modo a conceitualizar corretamente um carro e colocá-lo em funcionamento. **Relacionamentos:** podem ser implementados por meio do instanciamento de um objeto internamente a uma classe ou de um ponteiro para esse objeto.

Por exemplo, poderia ser declarada uma classe chamada **Valvula** e um objeto dessa classe ser instanciado dentro da classe **Motor**, estabelecendo-se, assim, um relacionamento entre motor e um de seus componentes internos: as válvulas. Veja o exemplo a seguir:

```
1 public class Motor {
2     private float cilindrada;
3     private int potencia;
4     private int temperatura;
5     private Valvula v; // instância da classe Valvula
6
7     Motor(int _cilindrada) {
8         cilindrada = _cilindrada;
9         temperatura = 0;
10        v = new Valvula();
11    }
12    public int getTemperatura() {
13        return temperatura;
14    }
15 }
```

Observe que somente os métodos da classe **Motor** podem acessar o objeto "v", uma vez que esse atributo é declarado como privado. Ainda, os atributos privados de **Valvula** também somente poderão ser acessados pelos métodos dessa classe. Por exemplo, imagine que a classe **Valvula** possui um atributo *booleano* e privado chamado "executa" e que inicia (no método construtor de **Valvula**) como false. Se algum método de Motor quiser atribuir **true** para o atributo "executa" de **Valvula**, então, terá que ser feito da seguinte forma:

```
1 v.setExecuta(true);
```

Obviamente, o método **setExecuta** (*boolean valor*) deverá estar implementado na classe **Valvula** e deverá ser público (public).



Video 02 - Classes e Objetos

Atividade 02

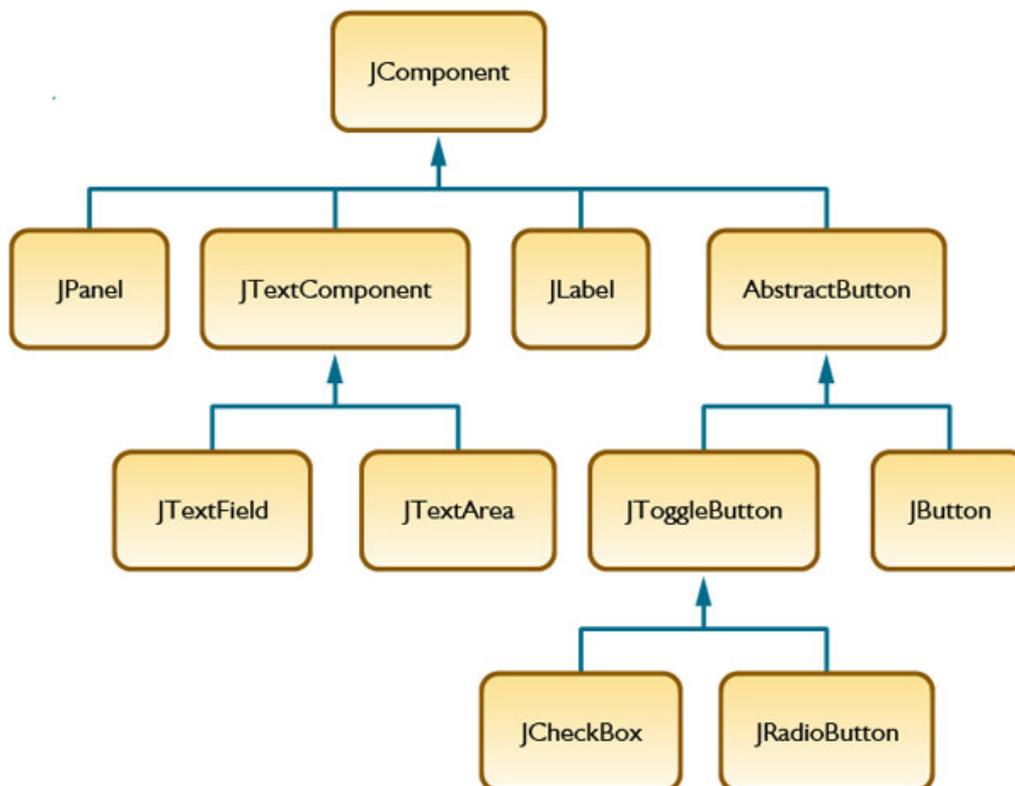
1. O que vem a ser entidades atômicas?
2. Como se devem acessar atributos privados (private)?

Herança

Herança é um princípio do paradigma de Programação Orientada a Objetos que permite criar uma classe a partir de outra já existente. Intuitivamente, o nome herança vem do fato de uma classe gerada a partir de outra conter (herdar) todos os atributos e métodos da superclasse (classe mãe). A ideia principal da herança é acrescentar novos atributos e métodos à classe filha, que vem a acrescentar-se aos já herdados. Tal característica serve para a diminuição de código duplicado.

É possível representar a hierarquia existente entre as classes através de uma imagem onde as classes herdeiras se posicionam abaixo das herdadas, indicando uma relação de derivação, assim como mostra a **Figura 1**.

Figura 01 - Parte da hierarquia do Swing



No caso da **Figura 1**, a classe *JComponent* é a classe mãe das classes *JPanel*, *JTextComponent*, *JLabel* e *AbstractButton*. Já as classes *JTextField* e *JTextArea* herdam de *JTextComponent*, e assim sucessivamente.

Em Java, a herança é implementada por meio da palavra-chave **extends**. Para indicar que determinada classe é uma subclasse, ou seja, que ela deriva (ou ainda, herda) de outra classe, dizemos que ela “estende” a classe mãe, como mostrado no exemplo a seguir:

```
1 public class MotorEletrico extends Motor {  
2     private float tensão_maxima;  
3     private float tensao_minima;  
4 }
```

No caso acima, declaramos uma classe MotorEletrico que herda os atributos e os métodos da classe mãe Motor, definida anteriormente nesta aula.



Vídeo 03 - Herança

Atividade 03

1. Para que serve a palavra-chave **extends** em Java?
2. Implemente uma hierarquia de classes representando os conceitos cadeira, cadeira de balanço e cadeira de rodas, onde as duas últimas são subclasses da primeira. Defina os atributos e métodos que achar apropriado.

Netbeans IDE

A partir de agora começaremos a segunda parte da aula. Vamos falar sobre um ambiente de desenvolvimento integrado que auxilia o programador nas tarefas básicas que compõe o desenvolvimento de um software.

O que é o NetBeans

O **NetBeans** é um ambiente de desenvolvimento multiplataforma que disponibiliza várias ferramentas de maneira integrada. Ele é utilizado pelos programadores para escrever, compilar, debugar (detectar erros no código) e instalar aplicações profissionais, sejam estas desktop, empresariais, para a web ou para dispositivos móveis (como celulares e tablets).

Esse tipo de ambiente de programação é conhecido como IDE (Integrated Development Environment – Ambiente de Desenvolvimento Integrado). No caso do Netbeans IDE, trata-se de um ambiente gratuito e de código aberto para desenvolvedores de software que utilizam as linguagens Java, C/C++, PHP, entre outras. Por ser multiplataforma, o NetBeans IDE pode ser executado em muitos sistemas operacionais, entre eles: Windows, Linux, Solaris e MacOS, por exemplo.

O **NetBeans** foi iniciado em 1996 por dois estudantes tchecos na Universidade de Charles, em Praga, quando a linguagem de programação Java ainda não era tão popular como atualmente. Primeiramente, o nome do projeto era Xelfi, em alusão ao Delphi, pois a pretensão deste projeto era ter funcionalidades semelhantes aos IDE's (ambientes de desenvolvimento integrado), então populares como o **Delphi** que eram mais atrativos por serem ferramentas visuais e mais fáceis de usar, porém com o intuito de ser totalmente desenvolvido em Java. Em 1999, o projeto já havia evoluído para um IDE proprietário, com o nome de **NetBeans DeveloperX2**, nome esse que veio da ideia da reutilização de componentes que eram a base do Java. Nessa época, a empresa [Sun Microsystems](#) havia desistido de seu IDE Java Workshop e, procurando por novas iniciativas, adquiriu o projeto **NetBeans DeveloperX2**, incorporando-o à sua linha de softwares.

Por alguns meses a Sun mudou o nome do projeto para **Forte for Java** e o manteve por um bom tempo como software proprietário; porém, em junho de 2000, a Sun disponibilizou o código fonte do **NetBeans IDE**, tornando-o uma plataforma Open Source. Desde então a comunidade de desenvolvedores que utilizam e contribuem com o projeto não parou de crescer, tornando-se um dos IDE's mais populares atualmente.

O **NetBeans** é totalmente escrito em Java, mas pode suportar o desenvolvimento em outras linguagens de programação, como o C, C++, Ruby ou o PHP. Também suporta linguagens de marcação como XML e HTML. Atualmente, está distribuído em diversos idiomas, inclusive o Português, e isso o tem tornado cada vez mais popular, facilitando o acesso a iniciantes em programação e possibilitado o desenvolvimento de aplicativos em diversas línguas.

Como o **NetBeans** é escrito em Java, ele é independente de plataforma, funcionando em qualquer sistema operacional que suporte a [máquina virtual](#) Java (JVM – ferramenta que executa o aplicativo em Java).

O NetBeans tem muitas vantagens e, além disso, é uma das mais poderosas ferramentas para o desenvolvimento de aplicações robustas e profissionais, fácil de usar e seu ambiente visual facilita a interação com o desenvolvedor, auxiliando-o na criação do código.

Requisitos de Hardware para a Instalação do NetBeans

Para instalar e trabalhar com o NetBeans de forma eficiente, você deverá possuir o seguinte equipamento mínimo: No sistema Windows, Windows XP (com Service Pack 2), Windows Vista, Windows 7 ou Windows 8:

- Processador Intel Pentium III 2.13 GHz (ou superior).
- 512 MB de memória RAM (é recomendado 1 GB).
- Resolução de vídeo 1024 x 768 e display de 32-bit.
- 400 MB de espaço disponível em disco.

No sistema Linux:

- Estação de trabalho Pentium III, 500 MHz.
- 512 MB de memória RAM (no mínimo).
- 415 MB de espaço livre (235 MBytes de espaço em disco permanente para o software e 180 MB de espaço em disco temporário necessário durante a instalação).

Download do NetBeans

Fazer o download de um programa significa baixar esse programa para o disco rígido do seu computador a partir do site do fabricante ou do desenvolvedor. Você também poderia baixar um programa qualquer através de sites de downloads públicos, como o Superdownloads ou o Baixaki, por exemplo, caso o programa desejado esteja disponível neles. Mas em alguns casos, como o nosso, é aconselhável baixar certos programas diretamente do site do fabricante. Além de ser mais seguro, encontra-se as últimas versões e atualizações. Então, vejamos como fazer o download da versão mais recente do NetBeans. Acompanhe-nos nessa tarefa.

1. Primeiramente, certifique-se de estar conectado à internet.
2. Em seguida, abra o seu browser (navegador), que pode ser o Firefox ou o Internet Explorer (os mais conhecidos e utilizados pelos internautas), utilizando os seus respectivos atalhos ou na área de trabalho ou no menu 'Iniciar'.
3. Na barra de endereço do browser, digite <netbeans.org> e pressione **Enter**.
4. Feito isso, será mostrada uma tela semelhante à da **Figura 2**:

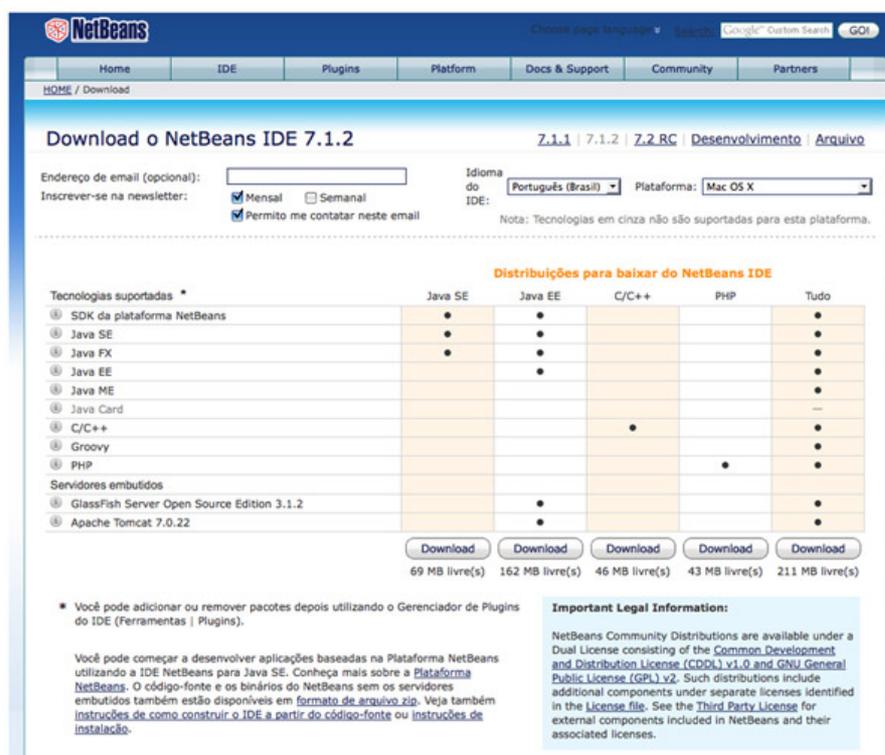
Figura 02 - Site oficial do NetBeans



Fonte: <netbeans.org>. Acesso em: 26 abr. 2012

- Você deve ter observado que o site se encontra em inglês. Mas não se preocupe. Se você quiser, poderá convertê-lo para português facilmente. Basta passar o mouse no link **choose page language**, localizado no topo do site, conforme mostra a **Figura 1**, que será mostrada uma lista de idiomas.
- Feito isso, clique na opção **Brazilian Portuguese**. Surgirá uma nova tela semelhante já convertida para português.
- Agora, clique no botão de cor laranja intitulado **'Download FREE'** para baixar o programa.
- Feito isso, surgirá a seguinte tela (**Figura 3**):

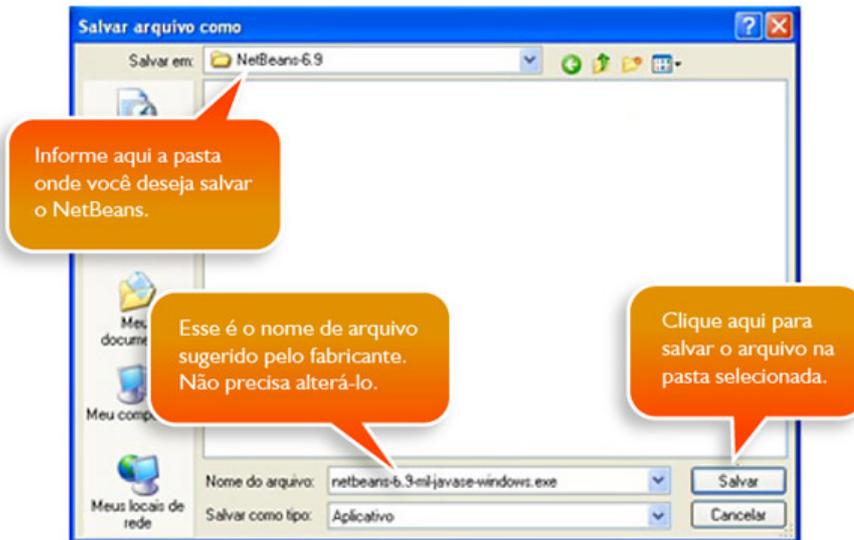
Figura 03 - Tela para a escolha da versão do NetBeans IDE



Fonte: <netbeans.org/>. Acesso em: 26 abr. 2012

- Agora clique no botão **'Download'** referente à primeira coluna para baixar a versão **'Java SE'** do NetBeans.
- Feito isso, será mostrada uma tela com uma pequena caixa de diálogo, clique no botão **'Salvar arquivo'**. Uma nova janela será mostrada para que você possa selecionar a pasta onde o arquivo deverá ser salvo, conforme ilustra a **Figura 4**.

Figura 04 - Janela para escolha da pasta onde será armazenado o NetBeans IDE



OBS.: Se no disco do seu computador não houver uma pasta específica para armazenar o NetBeans, crie uma nova com um nome sugestivo, compatível com o que você vai armazenar, de forma que você possa identificar, no futuro, qual o conteúdo daquela pasta. Veja nossa sugestão na figura acima. Quanto ao nome do arquivo, nesse caso, não é necessário alterá-lo.

11. Finalmente, clique no botão '**Salvar**' para salvar o arquivo na pasta selecionada.

Dependendo da sua conexão, o tempo de download desse arquivo poderá variar de alguns segundos a alguns minutos, mas, como o arquivo é pequeno, não demorará muito.

Terminado o processo de download do NetBeans, e considerando que não houve nenhum contratempo na realização dessa tarefa, passaremos agora à sua instalação. Da mesma forma que o processo anterior, você será orientado passo a passo, de forma que a instalação do NetBeans ocorra sem problemas.

Atividade 04

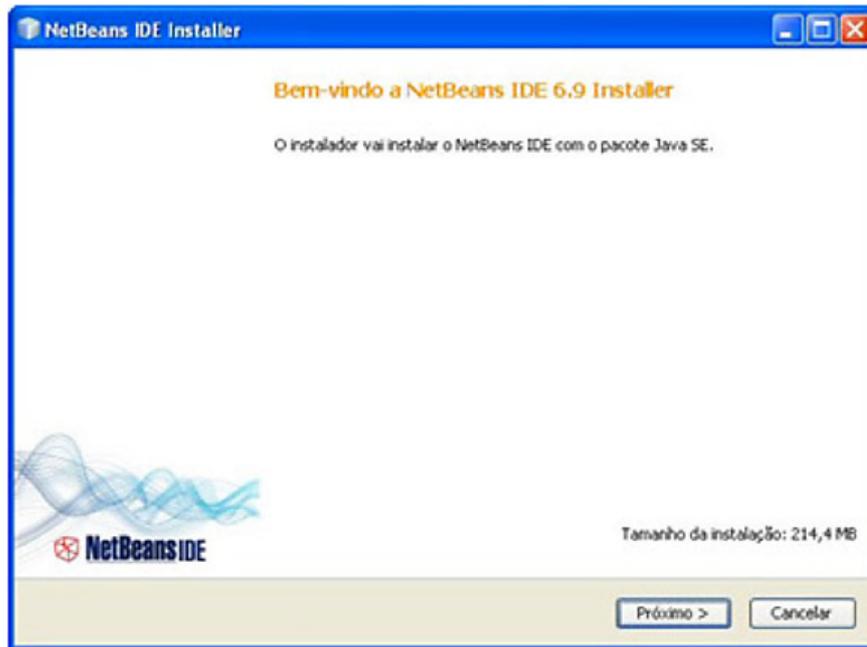
1. Faça uma descrição sucinta do que seja o NetBeans e para que ele serve.
2. O que você entende por 'baixar um programa'?
3. O que você entende por multiplataforma?
4. Para você, o que significa IDE?

Como Instalar o NetBeans

Considerando que o seu computador se encontra de acordo com as especificações exigidas, vamos lhe orientar passo a passo no processo de instalação dessa poderosa ferramenta de desenvolvimento, de forma que você não tenha nenhuma dificuldade nessa operação. Para isso, basta seguir atentamente as nossas orientações que você chegará lá. Vejamos, então, como fazer isso:

1. Inicialmente, abra o Windows Explorer e procure a pasta onde você salvou o NetBeans. Se você seguiu nossa orientação, sua pasta é o próprio nome do programa: **NetBeans**.
2. Clique nessa pasta para abri-la. O arquivo do NetBeans deverá aparecer.
3. Para executar o arquivo, dê um duplo clique sobre o seu nome (ou clique no arquivo e pressione **Enter** caso você tenha dificuldade com o duplo clique).
4. Feito isso, será mostrada uma caixa de diálogo.
5. Clique no botão '**Executar**' para iniciar a instalação. Uma pequena caixa de diálogo aparecerá muito rapidamente somente para configurar o instalador.
6. Em seguida, será mostrada a janela principal para iniciar a instalação propriamente dita do NetBeans, conforme ilustra a **Figura 5**.

Figura 05 - Janela principal para a instalação do NetBeans



7. Clique no botão '**Próximo**' para dar início à instalação.
8. A janela seguinte mostra os termos do **Contrato de Licença** para o uso do programa.
9. Basta marcar o check box para aceitar os termos do contrato. Do contrário, se você não marcar essa opção, a instalação não se concretizará e a operação será abortada.
10. Depois disso, clique no botão '**Próximo**' para continuar com a instalação.
11. Nessa nova janela, o instalador solicita as pastas onde o programa e suas bibliotecas deverão ser armazenadas. Para o nosso caso, não altere nenhuma das pastas sugeridas pelo instalador. Apenas clique no botão '**Próximo**' para continuar.
12. Na próxima janela, apenas a título de informação, o instalador confirma a pasta na qual o NetBeans será instalado e o espaço que será ocupado em seu disco quando toda a instalação estiver completada.
13. Clique no botão '**Instalar**' para iniciar a instalação propriamente dita.
14. Será mostrada uma nova janela com uma barra de progresso para que você possa acompanhar o processo da instalação. O tempo de instalação vai depender dos recursos da sua máquina, mas, de qualquer forma, não durará mais do que alguns segundos.

15. Terminado o processo de instalação, uma nova janela será mostrada (ver **Figura 6**):

Figura 06 - Janela para confirmação de conclusão da instalação



16. Clique no botão '**Finalizar**' para encerrar a instalação do NetBeans IDE.

17. Feito isso, você será redirecionado para uma página para registrar o NetBeans. Não é necessário registrar o programa para poder utilizá-lo. Pode fechar a página, se quiser. A operação de instalação está encerrada.

Partiremos agora para a próxima etapa, que é a sua execução.

Como Executar o NetBeans

Após a instalação, o NetBeans IDE deve ter criado um ícone na área de trabalho, como também um atalho no menu '**Iniciar**'. Para executá-lo, você poderá utilizar qualquer uma das opções. Para facilitar as coisas, iremos executá-lo a partir do ícone na área de trabalho. Para isso:

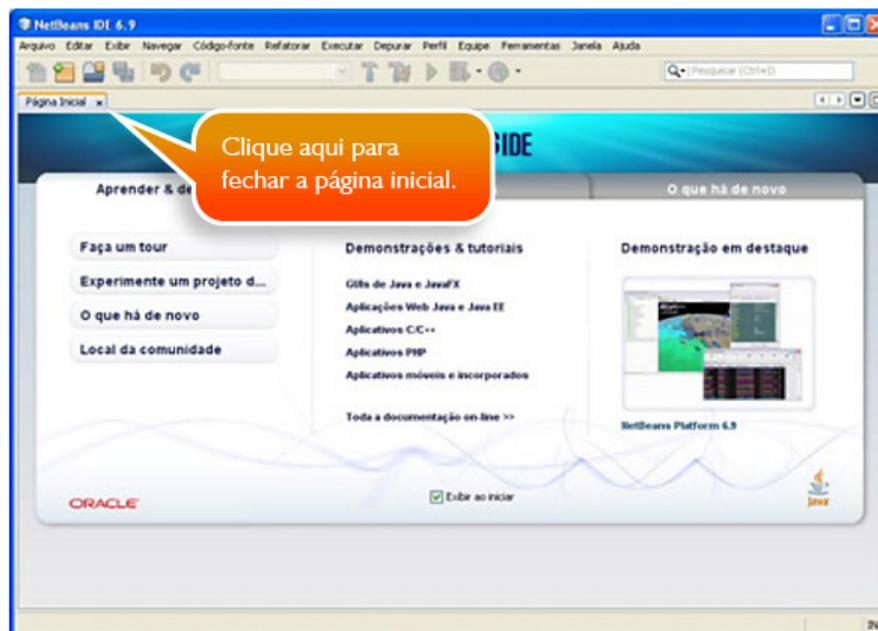
1. Acesse a área de trabalho utilizando as teclas '**Windows + D**' simultaneamente.
2. Procure o ícone do NetBeans, conforme mostra a **Figura 7**, e dê um duplo clique nele.

Figura 07 - Atalho do NetBeans na área de trabalho



3. Será mostrada uma tela de inicialização do NetBeans somente para carregar alguns módulos da ferramenta.
4. Depois disso, será mostrada a tela principal do NetBeans, e sobre ela uma outra janela para o registro do NetBeans. Por enquanto, não se preocupe com essa janela. Apesar do registro ser gratuito, você poderá fazer isso quando quiser. Passe à frente clicando no botão '**Lembre-me mais tarde**'.
5. A **Figura 8** a seguir mostra a tela principal do NetBeans:

Figura 08 - Janela principal do NetBeans



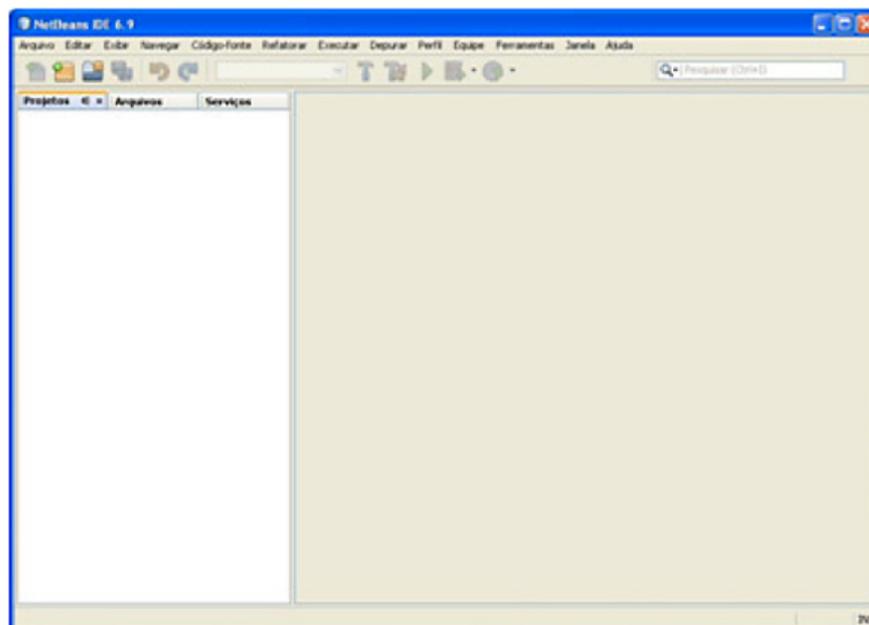
6. Nessa página principal, você encontrará três abas distintas:

- Aprender & descobrir.
- Meu NetBeans.
- O que há de novo.

Quando você tiver mais experiência, poderá utilizar os vários recursos disponíveis no NetBeans através dessas abas. A maioria desses recursos são links para várias páginas, e grande parte está em Inglês. Por enquanto, vamos fechar essa página e conhecer o ambiente de desenvolvimento do NetBeans. Isso é o que nos interessa.

7. Para isso, clique no pequeno 'x' ao lado de '**Página Inicial**', conforme mostrado na **Figura 8**.
8. Agora será mostrada uma janela com o ambiente de desenvolvimento do NetBeans. Como não temos ainda nenhum projeto criado, essa tela será mostrada como segue (**Figura 9**):

Figura 09 - Janela do ambiente de desenvolvimento do NetBeans

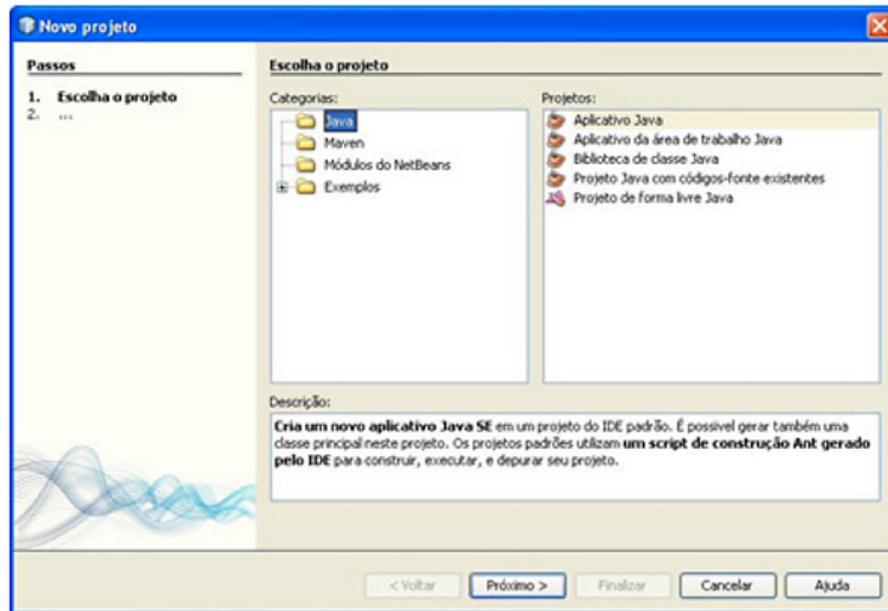


9. Para que possamos conhecer os recursos desse ambiente, precisamos criar um novo projeto. É exatamente isso que faremos logo a seguir. Acompanhe-nos.

Explorando as Funcionalidades do NetBeans

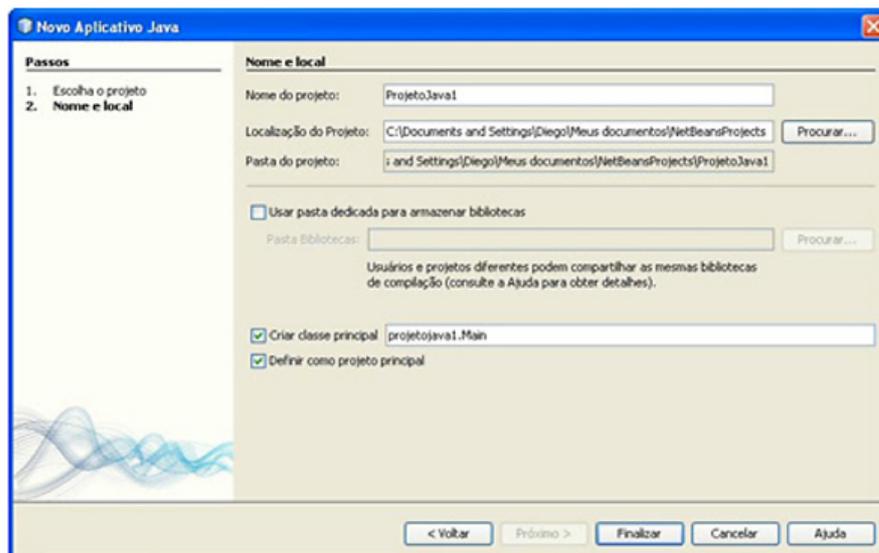
Vamos agora explorar o NetBeans e, assim, desenvolver as nossas aplicações de forma eficiente e produtiva. Agora criaremos um pequeno projeto, bem simples por sinal, apenas para mostrar uma frase na tela, utilizando o NetBeans. Vejamos como fazer isso:

1. No menu principal, clique em 'Arquivo' e, em seguida, **'Novo projeto'**.
2. Será apresentada uma nova janela, conforme mostra a **Figura 10** a seguir:
Figura 10 - Janela para seleção da categoria e tipo de projeto



3. Nesta janela, selecione a categoria do projeto, no nosso caso **'Java'**, e em seguida o tipo de projeto, **'Aplicativo Java'**. Clique no botão **'Próximo'** para prosseguir.
4. Na janela seguinte (**Figura 11**) altere apenas o nome do projeto de **'JavaApplication1'** para **'ProjetoJava1'**.

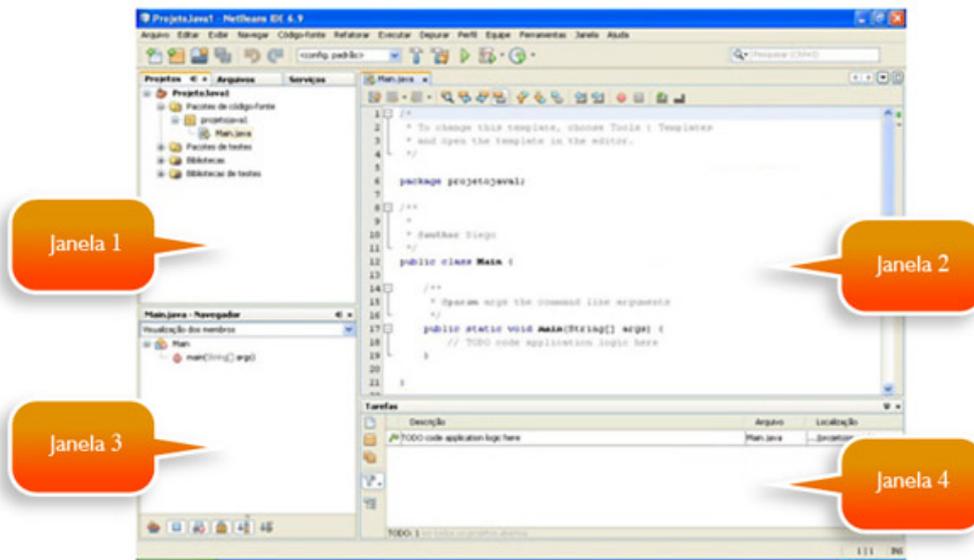
Figura 11 - Janela para informação do nome do novo projeto e local de armazenamento



5. Clique no botão **'Finalizar'** para encerrar a criação do projeto.

6. Feito isso, e alguns segundos depois de processar a criação do novo projeto, será mostrada a seguinte tela (**Figura 12**):

Figura 12 - Janela após a criação de um novo projeto



Você deve ter observado que o ambiente do NetBeans foi dividido em quatro áreas, que chamaremos de janelas, para facilitar.

A Primeira Janela

Essa primeira janela possui três abas. Vamos conhecê-las?

1. A primeira aba, chamada '**Projetos**', mostra as pastas e bibliotecas necessárias para o desenvolvimento do projeto recém-criado, com destaque para o arquivo principal do projeto, o '**Main.java**'. Veja na **Figura 13**.

Figura 13 - Aba Projetos mostrando a criação do projeto e suas pastas de arquivos e bibliotecas



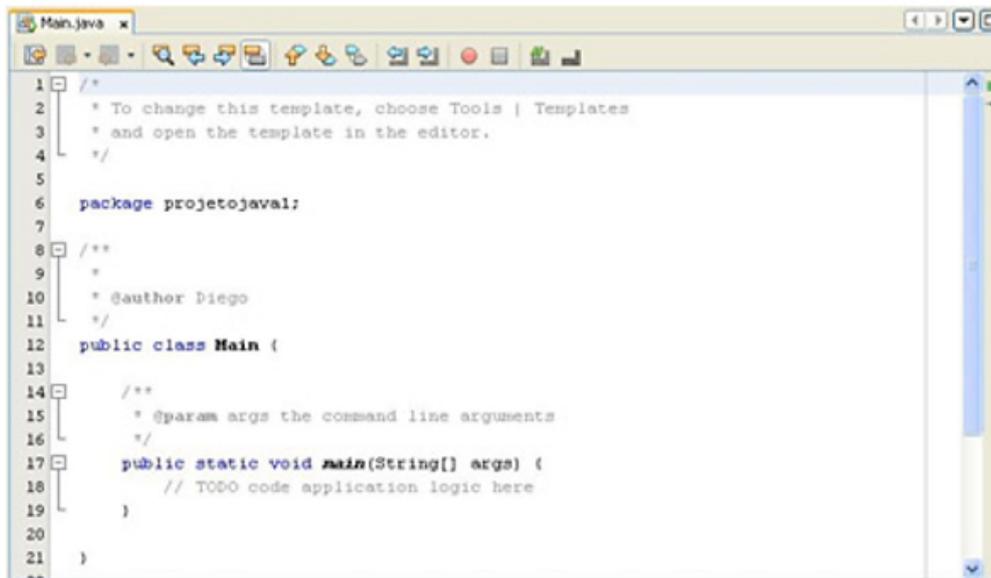
2. A segunda aba, chamada '**Arquivos**', mostra as pastas e os arquivos necessários para o projeto.
3. A terceira aba, chamada '**Serviços**', mostra alguns recursos que somente serão utilizados em projetos futuros. Por isso não nos preocuparemos com ela agora.

Todas essas pastas, arquivos e serviços são criados automaticamente pelo NetBeans toda vez que criarmos um novo projeto.

A Segunda Janela

Chamada de '**Main.java**', é a janela do código propriamente dito. É como se fosse um editor de textos do tipo bloco de notas. Só que esse é específico para a criação de projetos em Java, pois ele dispõe de recursos que os editores de textos convencionais não possuem. Esse código que você vê foi criado automaticamente pelo NetBeans. Qualquer projeto que você criar terá pelo menos essas linhas de código. Veja essa janela em destaque na **Figura 14**, além de algumas características dela.

Figura 14 - Janela para criação e edição de código



Observe que todas as linhas são numeradas. Isso facilita a procura por possíveis erros que possam ser detectados no código durante a compilação e execução do projeto. Você fica sabendo exatamente onde ocorreu o erro, o que é comum acontecer. Observe também a diferença de cores entre as diversas partes do código, tais como nomes de classes, variáveis, funções etc. Isso facilita muito quando você estiver digitando o seu algoritmo. Você ficará sabendo de imediato se o seu comando está certo ou não de acordo com a cor que foi configurada para ele. Essas cores podem ser modificadas de acordo com as suas necessidades. Posteriormente falaremos desse assunto.

Os pequenos quadrados com um sinal de subtração (-) em seu interior (linhas 1, 8, 14 e 17) são utilizados para esconder/mostrar o código de acordo com sua necessidade, principalmente se o projeto tiver uma grande quantidade de linhas. Quando você clica em um quadrado desses, o sinal de subtração é alterado para um sinal de adição (+), informando que o conteúdo se encontra oculto.

Alguns dos ícones localizados acima do editor são utilizados para navegar dentro do código, outros não são importantes no momento. Você conhecerá a utilidade deles gradativamente, à medida que forem sendo necessários.

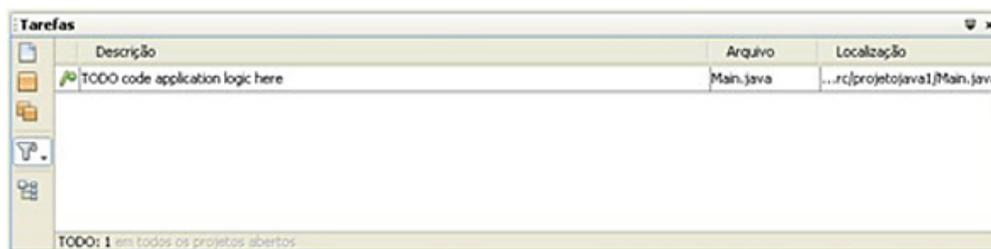
A Terceira Janela

Essa janela, chamada de '**Navegador**', será detalhada em futuros projetos dessa disciplina.

A Quarta Janela

Essa janela, chamada de '**Tarefas**', conforme mostra a **Figura 15**, mostra a descrição do seu projeto e sua localização no disco rígido. Os ícones localizados à sua esquerda serão estudados de acordo com os projetos criados. Nosso projeto é muito pequeno para utilizá-los agora.

Figura 15 - Janela mostrando nome e localização do projeto



Essa janela poderá apresentar também uma outra aba chamada de 'Saída', que aparecerá quando você executar a aplicação. Ela mostrará informações sobre a compilação do projeto, se deu certo ou não, o tempo de execução, ou os possíveis erros, se for o caso. Veremos isso mais adiante quando executarmos nosso pequeno projeto.

Atividade 05

1. Qual a utilidade das cores utilizadas no editor de códigos do NetBeans?
2. Para que servem os pequenos quadrados que aparecem ao lado de algumas linhas de código?

O Primeiro Projeto no NetBeans

Nesse nosso primeiro projeto criaremos um programa que imprime uma pequena frase na tela. Escreveremo-la diretamente no código. Veja como é simples:

1. Na janela de código, posicione o cursor no final da linha **18** e tecle '**Enter**'. Isso fará com que uma linha em branco seja criada logo abaixo.
2. Nessa linha recém-criada, digite a seguinte declaração:

```
1 System.out.println ("Bem-vindo ao Mundo da Programação Java!");
```

Acreditamos que o comando utilizado não seja novidade para você, portanto, não será necessário tecer comentários sobre ele. Você já deve ter digitado esse comando dezenas de vezes durante o curso.

3. Seu código deverá ficar igual ao mostrado na **Figura 16**.

Figura 16 - Como deverá ficar o seu código após a inserção da declaração

```
14  /**
15   * @param args the command line arguments
16   */
17  public static void main(String[] args) {
18      // TODO code application logic here
19      System.out.println ("Bem-vindo ao Mundo da Programação Java!");
20  }
21
```

Observou a diferença de cores entre os diversos comandos? É pelas cores que você identifica quem é quem no código, e com isso você poderá detectar os erros até mesmo durante a digitação.

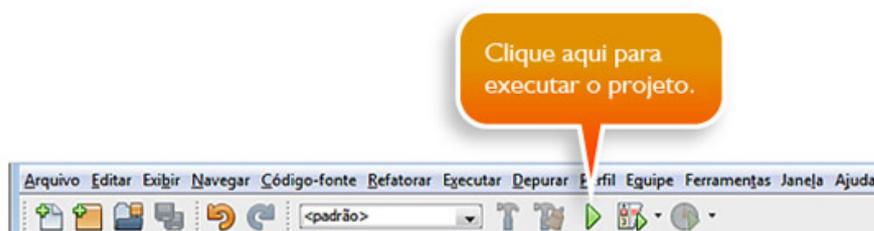
4. Esse código, após executado, deverá mostrar apenas a seguinte expressão (ou string):

Bem-vindo ao Mundo da Programação Java

Para executar a aplicação, utilize um dos seguintes recursos:

5. Clicando no ícone '**Executar Main Project**', conforme mostra a **Figura 17**.

Figura 17 - Barra de ferramentas para execução da aplicação.



6. Ou simplesmente pressionando a tecla **F6**.

7. Feito isso, será mostrado na aba '**Saída**' da quarta janela do IDE do NetBeans, conforme falamos anteriormente, o resultado da execução. Se tudo tiver corrido bem, sem nenhum erro de digitação ou de sintaxe, o resultado será o seguinte (**Figura 18**):

Figura 18 - Resultado da execução da aplicação



Nessa janela estão sendo mostrados:

- O resultado da compilação, ou seja, a string de saída (**linha 2**).
- A declaração de ausência de erros (**linha 3**).
- E o tempo gasto na execução (**linha 3**).



Vídeo 04 - Primeiro Projeto Netbeans

Atividade 06

1. Qual a tecla utilizada para executar uma aplicação no NetBeans?
2. Crie um novo projeto utilizando a mesma declaração do nosso exemplo, de forma que a expressão seja impressa em diversas linhas, no seguinte formato:

```
Bem-vindo  
ao Mundo  
da Programação
```

```
Java!
```

Conclusão

Bem, chegamos ao final de nossa aula referente à introdução ao NetBeans. Esperamos que você tenha absorvido o conhecimento necessário para desenvolver suas primeiras aplicações utilizando essa plataforma. Nas próximas aulas, você saberá como utilizar gradativamente os seus recursos. Não procure saber tudo antes do tempo. Boa sorte!

Resumo

Nesta aula, você relembrou e reavaliou os principais conceitos da OO, os quais serão fundamentais para o restante da nossa disciplina, pois as classes da linguagem de programação Java, de maneira geral, se apoiam fortemente nesses conceitos e, a partir de agora, vai ser comum a interação com herança, métodos e atributos públicos e privados, entre outros conceitos que serão definidos ao longo da disciplina.

Além disso, você conheceu a plataforma de desenvolvimento NetBeans, aprendeu como baixá-lo e instalá-lo corretamente, ficou familiarizado com sua interface e conheceu alguns dos seus recursos mais utilizados. Nesse primeiro contato com o NetBeans, você criou um primeiro projeto chamado '**ProjetoJava1**', diretamente na área de código, para imprimir um texto de apenas uma linha. A partir da próxima aula, você passará a utilizar o NetBeans para a criação de aplicativos com uma GUI, ou seja, utilizando os componentes do Swing para criar projetos com visual mais profissional.

Autoavaliação

1. Dê uma definição resumida de POO, de acordo com os conceitos estudados.
2. Qual o padrão ideal para nomear variáveis?
3. A afirmação: "Atributos e métodos privados somente podem ser acessados por métodos da própria classe." é verdadeira ou falsa? Justifique sua resposta.
4. Quais as vantagens de se utilizar o NetBeans em comparação aos editores de texto convencionais? Justifique.
5. No ambiente do NetBeans, onde é mostrado o resultado da aplicação após ser executada?

6. Faça um programa utilizando duas variáveis, 'a' e 'b', com dois valores quaisquer, de forma que o resultado seja o produto delas com uma declaração impressa no seguinte formato:

O produto de 'a' vezes 'b' é igual a:

7. Crie um novo projeto utilizando um loop 'for' entre 1 e 15, de forma que sejam impressos apenas os números ímpares.

Referências

CHAPTER 13: Inheritance. Chapter1 3:em: <<http://www.iro.umontreal.ca/~pift1025/bigjava/Ch13/ch13.html>>. Acesso em: 17 abr. 2012.

DEITEL, Harvey M.; DEITEL, Paul J. **Java**: como programar. 6. ed. São Paulo: Prentice-Hall, 2005. 1152p.

NETBEANS. Disponível em: <<http://netbeans.org>>. Acesso em: 26 abr. 2012.

ORACLE. **Instalação da plataforma:** Windows. Disponível em: <http://java.sun.com/javase/pt_BR/6/webnotes/install/install_jdk1.6_3-nb5.5.1_ml.html#windows>. Acesso em: 26 abr. 2012a.

_____. **Instalação da plataforma:** Linux. Disponível em: <http://java.sun.com/javase/pt_BR/6/webnotes/install/install_jdk1.6_3-nb5.5.1_ml.html#linux>. Acesso em: 26 abr. 2012b.

ORACLE BRASIL. Disponível em: <<http://www.sun.com.br>>. Acesso em: 26 abr. 2012.