

Introdução ao IDE Netbeans (Programação Java)

Conteúdo

1.	Introdução.....	1
1.1.	Programas necessários.....	1
2.	Visão geral da criação da primeira aplicação usando o NetBeans.....	1
3.	Criação da primeira aplicação usando o NetBeans	2
3.1.	Passos para a criação de um Projeto no NetBeans.....	2
3.2.	Adicionar código no arquivo fonte gerado ('Generated Source File').....	3
3.3.	Compilar o arquivo fonte gerando o arquivo com extensão 'class'.....	5
3.4.	Executar o Programa.....	5
4.	Exercícios	6
5.	Exercícios para casa.....	6

1. Introdução

Os programas em geral são desenvolvidos com a ajuda dos chamados Ambientes Integrados de Desenvolvimento, ou simplesmente IDE (Integrated Development Environment). Estes ambientes fornecem recursos importantes para auxiliar e facilitar o trabalho do programador, como a identificação de palavras reservadas durante a escrita do código fonte, identificação de erros de sintaxe, depuração¹ e compilação.

O IDE adotado no curso é o NetBeans (<http://www.netbeans.org/products/ide/>), particularmente, a versão 5.0, usa o Java 5 (com o JDK 1.5). No NetBeans, como em outros IDEs, todo o desenvolvimento de aplicativos (ou programas) é realizado dentro de um projeto².

Na seqüência deste texto é apresentada uma introdução à utilização do IDE do software adotado no desenvolvimento dos programas (aplicativos) em Java.

1.1. Programas necessários

Para executar esta aula prática, é necessário ter instalado o NetBeans e o JDK:

- O IDE NetBeans versão 5.0 ou superior, que pode ser obtido a partir de <http://java.sun.com/javase/downloads/> ou <http://www.netbeans.org/downloads/index.html>. Quando se instala o NetBeans, o JDK é instalado automaticamente.
- O Java SE Development Kit versão 5 (JDK 5), ou superior, que pode ser obtido a partir de <http://java.sun.com/javase/downloads/> (escolher JDK 5.0 Update 9 ou superior), caso uma versão diferente do JDK seja desejada. É possível ter várias versões do JDK instaladas e configurar o NetBeans para utilizar a mais adequada a cada situação.

Nota: As telas capturadas e mostradas a seguir correspondem ao IDE NetBeans 5.0, que usa o JDK 1.5.0.

2. Visão geral da criação da primeira aplicação usando o NetBeans

Esta primeira aplicação, chamada BemVindo.java, simplesmente mostrará a frase "Sejam bem vindos ao Java!". Para criar este programa é necessário:

¹ Processo de identificação e remoção de erros, facilitado pelos IDEs.

² Um projeto consiste num grupo de arquivos com código fonte em Java, bem como informações necessárias para construir e executar este projeto.

1) Criar um projeto IDE ('IDE project')

Ao criar um projeto IDE, cria-se o ambiente necessário para construir e executar aplicações. O uso de projetos IDE elimina problemas normalmente associados com o compilador javac e demais ferramentas. Para construir e executar aplicações basta escolher um menu dentro do IDE.

2) Adicionar código para o arquivo fonte gerado

Um arquivo fonte contém código, escrito na linguagem de programação JAVA. Como parte da criação de um projeto IDE, é gerado automaticamente o esqueleto de um arquivo fonte, que deverá ser modificado ao adicionar-se a mensagem "Sejam bem vindos ao Java".

3) Compilar o arquivo fonte gerando o arquivo com extensão 'class'

O IDE inicia o compilador (javac), que traduz o arquivo fonte em instruções que podem ser traduzidas pelo JVM ('JAVA virtual machine'). Estas instruções são conhecidas como 'bytecodes'.

4) Executar o programa

A aplicação poderá ser executada por meio do IDE.

3. Criação da primeira aplicação usando o NetBeans

A primeira aplicação usando o Netbeans irá mostrar uma mensagem de boas vindas ao Java na tela.

3.1. Passos para a criação de um Projeto no NetBeans

- 1) Iniciar o NetBeans (clicar 2X no ícone NetBeans 5.0).
- 2) No IDE NetBeans, escolher **File | New Project** como mostrado na Figura 1.

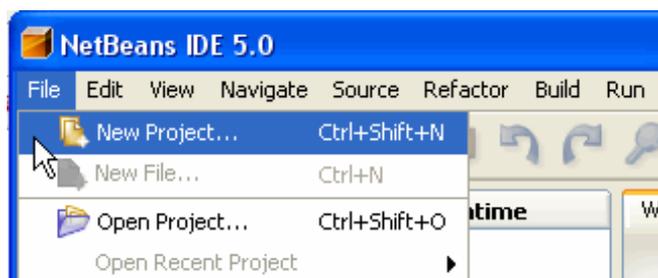


Figura 1 - NetBeans com o item de menu **File | New Project** selecionado.

- 3) Na janela 'New Project', expandir a categoria 'General' e selecionar 'Java Application', como mostrado na Figura 2:

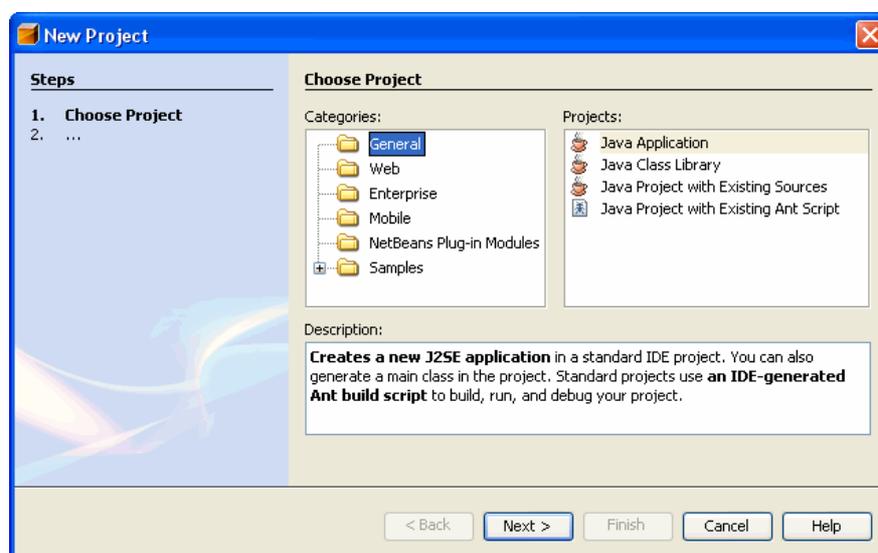


Figura 2 - Janela 'New Project' mostrando categoria 'General' e tipo de projeto 'Java Application'.

- 4) Clicar no botão 'Next>'
- 5) Preencher os campos da janela 'New Java Application', como mostrado na Figura 3:
 - No campo 'Project Name', digitar 'Bem Vindo ao Java'
 - No campo 'Create Main Class', digitar 'bemvindoaojava.BemVindo'
 - Manter selecionada a caixa de seleção 'Set as Main Project'

Nota: O campo 'Project Location:' deverá ser modificado com a localização preferida para a pasta contendo o projeto em desenvolvimento.

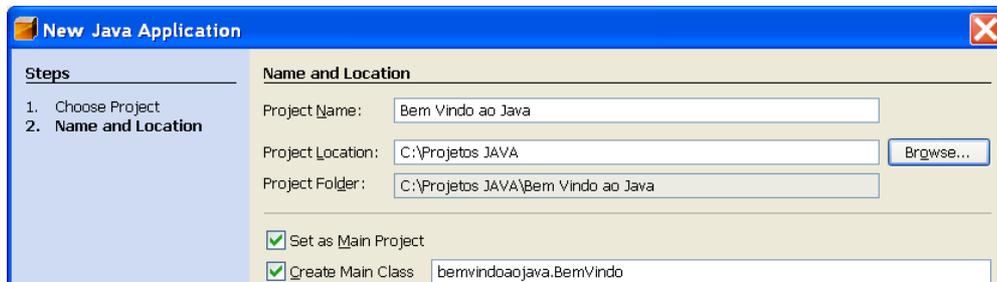


Figura 3 - NetBeans mostrando a página 'Name and Location'.

- 6) Clicar no botão 'Finish'.
- 7) Clicar no botão 'Next>'

O projeto é criado automaticamente e aberto no IDE. Algumas das janelas abertas no IDE são apresentadas na Figura 4 e detalhadas a seguir.

- A janela 'Projects', que contém uma visualização em árvore dos componentes do projeto, incluindo arquivos fonte, bibliotecas relacionadas com o código em construção, etc.
- A janela 'Source Editor' com um arquivo chamado 'BemVindo.java' aberto.
- A janela 'Navigator', que pode ser usada para a navegação rápida entre elementos internos a uma classe selecionada.

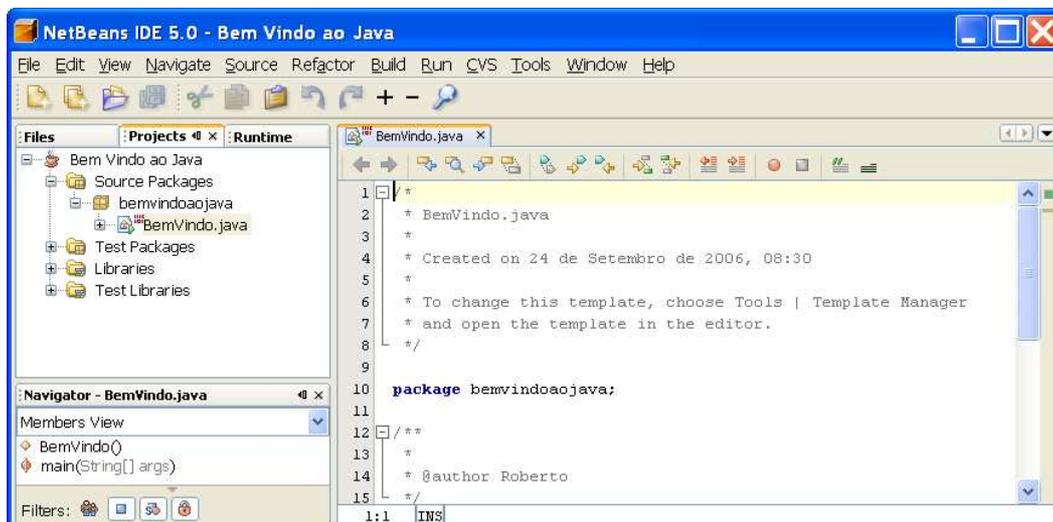


Figura 4 - IDE NetBeans com o projeto 'Bem Vindo ao Java' aberto.

3.2. Adicionar código no arquivo fonte gerado ('Generated Source File')

- 8) Como a caixa de seleção 'Create Main Class' permaneceu selecionada (passo 5 do item 3.1) o IDE criou o esqueleto de uma classe. Portanto, pode-se adicionar a mensagem "Sejam bem vindos ao Java!" a este esqueleto de código apenas substituindo a linha de código:

```
// TODO code application logic here
```

pelas linhas:

```
System.out.println("\n*****");
System.out.println("Sejam bem-vindos ao Java!");
System.out.println("*****\n");
```

Notas:

- a) Estas quatro linhas de código acima são comentários e não afetam o modo como o programa será executado.
- b) Cuidado ao digitar o código, comandos e nomes de arquivos. A linguagem Java é sensível a letras maiúsculas (caixa alta ou “upper case”) e a letras minúsculas (caixa baixa ou “lower case”). Assim:

BemVindos  bemvindos

- 9) Toda mudança feita no código é salva escolhendo **File | Save** do menu principal ou apenas teclando **Ctrl + S**. Salve o seu projeto.

- 10) O código apresentado na janela BemVindo.Java do IDE NetBeans deverá ser parecido com o código a seguir:

```
/* BemVindo.java
 * Created on 24 de Setembro de 2006, 08:30
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */
package bemvindojava;
/**
 *
 * @author Roberto
 */
public class BemVindo {
    /** Creates a new instance of BemVindo */
    public BemVindo() {
    }
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        System.out.println("\n*****");
        System.out.println("Sejam bem-vindos ao Java!");
        System.out.println("*****\n");
    }
}
```

- 11) É possível também apagar partes do código que não serão aproveitadas nesta parte do curso de tal forma que o código final fique como a seguir.

```
/* BemVindo.java
 * Created on 24 de Setembro de 2006, 08:30
 */
public class BemVindo {
    public static void main(String[] args) {
        System.out.println("\n*****");
        System.out.println("Sejam bem-vindos ao Java!");
        System.out.println("*****\n");
    }
}
```

3.3. Compilar o arquivo fonte gerando o arquivo com extensão 'class'

12) Para compilar o arquivo fonte, escolha **Build | Build Main Project** do menu principal do IDE, ou apenas tecle F11. A janela 'Output' apresenta uma saída similar á mostrada na Figura 5.

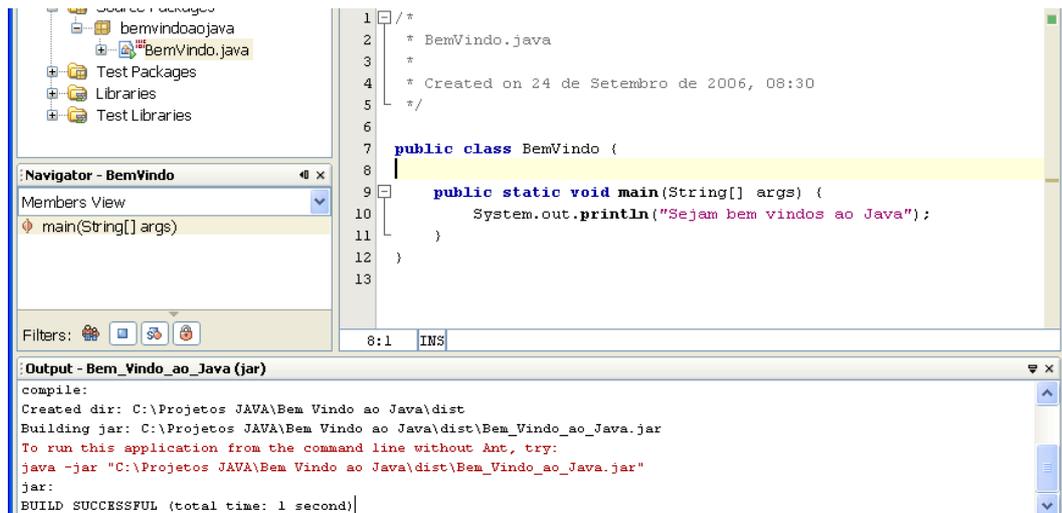


Figura 5 - Janela 'Output' mostrando os resultados do projeto Bem Vindo ao Java.

13) O programa terá sua compilação e execução bem sucedida se a saída terminar com a sentença **BUILD SUCCESSFUL**. Porém, se ocorrer erro, como o de sintaxe, a saída terminará com a sentença **BUILD FAILED**.

14) Ao construir o projeto, é gerado o arquivo **BemVindo.class**, formado por 'bytecodes'. Para verificar a localização deste arquivo basta abrir a janela 'Files' e expandi-la (**Bem Vindo ao Java/build/classes/bemvindoaojava/ BemVindo.class**) como mostrado na Figura 6.

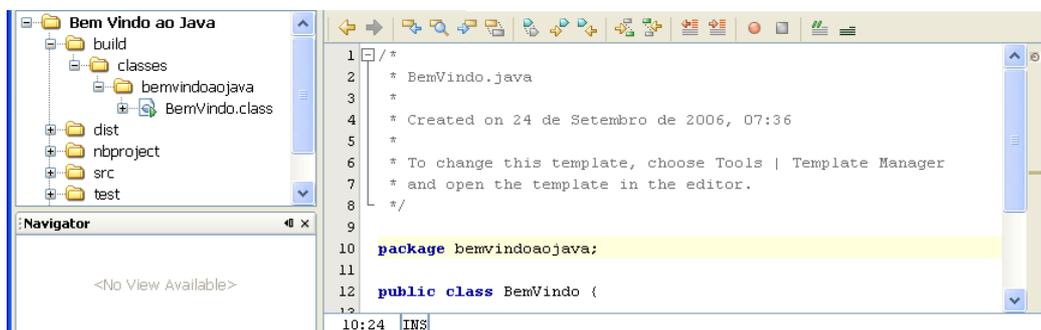


Figura 6 - Janela 'Files' mostrando o arquivo .class gerado após compilação.

15) Após a construção do projeto, basta executar o programa como explicado a seguir.

3.4. Executar o Programa

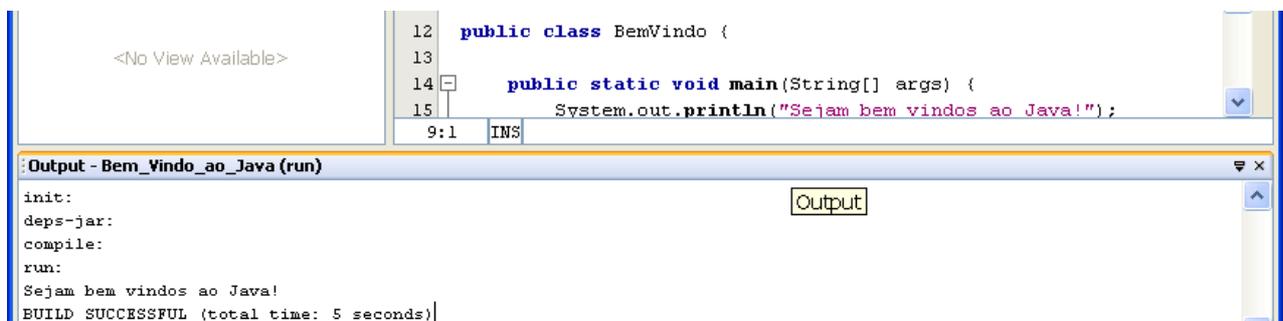


Figura 7 - O programa imprime a mensagem "Sejam bem vindos ao Java!" na janela 'Output'.

16) No menu principal do IDE escolher **Run | Run Main Project** ou tecle F6. A Figura 7 mostra a saída do programa.

4. Exercícios

17) Escrever um programa em Java que leia dois números e imprima a soma deles. Faça com que o projeto se pareça com o programa se pareça com o código a seguir:

```
package soma;
import java.util.*;
public class Soma {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("digite um numero: ");
        int NumA = sc.nextInt();
        System.out.println("digite outro numero: ");
        int NumB = sc.nextInt();
        int Soma = NumA + NumB;
        System.out.println ("Soma dos numeros: " + Soma);
    }
}
```

18) Altere o programa para que ele também calcule subtração, multiplicação e divisão.

5. Exercícios para casa

- 1) Reescrever e executar os programas propostos na nota de aula 'Primeiro programa em Java (compilação e execução)' usando o IDE Netbeans.
- 2) O programa a seguir tem um erro. Encontre o erro e corrija-o de tal forma que possa ser compilado e executado corretamente. Qual é o erro?

```
// INTENTIONALLY UNCOMPILABLE!

/**
 * The HelloWorldApp class implements an application that
 * simply prints "Hello World!" to standard output.
 */
class HelloWorldApp2 {
    public static void main(String[] args) {
        System.out.println("Hello World!"); // Display the string.
    }
}
```

- 3) Modifique o aplicativo BemVindo.java de tal forma que exiba cada palavra da mensagem 'Sejam bem vindos ao Java' numa linha diferente.
- 4) Quando você compila um programa escrito em Java, o compilador converte o código fonte que pode ser entendido por um leitor humano em um código independente da plataforma, que a JVM pode entender. Como é chamado este código independente da plataforma?
- 5) Qual das seguintes alternativas contém um comentário não válido?
 - a. `/** comment */`
 - b. `/* comment */`
 - c. `/* comment`
 - d. `// comment`
- 6) Ao declarar o método **main**, qual o modificador deverá vir primeiro, **public** ou **static**?
- 7) Quais parâmetros o método **main** define?
- 8) Qual a correta sintaxe empregada ao declarar o método **main**?