

Java-Lab1

Notas de Aula
Prof. André Bernardi
andrebernardi@unifei.edu.br





ECOP15 - Plano de curso (P1 e P2)

Início	Fim	Descrição
30/08/2021	06/09/2021	Revisão
13/09/2021	13/09/2021	Manipulação de Strings
20/09/2021	20/09/2021	Desenho e Java 2D
27/09/2021	27/09/2021	Manipulação de Arrays e Desenho
04/10/2021	04/10/2021	Interface Gráfica - Componentes
11/10/2021	18/10/2021	Interface Gráfica - Layout e Painéis
25/10/2021	25/10/2021	Interface Gráfica - Janela e Menus
01/11/2021	01/11/2021	Processamento de Exceções
08/11/2021	08/11/2021	Primeira Avaliação
22/11/2021	22/11/2021	Janelas Internas e Subclasse de JPanel
29/11/2021	29/11/2021	Persistencia de dados - Entrada e Saida em Arquivos
06/12/2021	06/12/2021	Multimidia e Threads
13/12/2021	13/12/2021	Multimidia e Manipulação de Imagens
13/12/2021	13/12/2021	Segunda Avaliação



ECOP15 - Plano de curso (P3 e P4)

Início	Fim	Descrição
31/08/2021	31/08/2021	Revisão
14/09/2021	14/09/2021	Manipulação de Strings
21/09/2021	21/09/2021	Desenho e Java 2D
28/09/2021	28/09/2021	Manipulação de Arrays e Desenho
05/10/2021	05/10/2021	Interface Gráfica - Componentes
19/10/2021	19/10/2021	Interface Gráfica - Layout e Painéis
26/10/2021	26/10/2021	Interface Gráfica - Janela e Menus
09/11/2021	09/11/2021	Primeira Avaliação
16/11/2021	16/11/2021	Processamento de Exceções
23/11/2021	23/11/2021	Janelas Internas e Subclasse de JPanel
30/11/2021	30/11/2021	Persistencia de dados - Entrada e Saida em Arquivos
07/12/2021	07/12/2021	Multimidia e Threads
14/12/2021	14/12/2021	Segunda Avaliação
14/12/2021	14/12/2021	Multimidia e Manipulação de Imagens



Avaliações

- Primeiro Bimestre:
 - Labs de 01 a 07 – 60%
 - 08 e 09 de novembro – 40%
- Segundo Bimestre:
 - Labs de 08 a 11 – 60%
 - 13 e 14 de dezembro – 40%



Referências

- Java How to Program 3, 4, 5, 6, 7, 8, 9, 10 ed. - Paul Deitel and Harvey Deitel.
- Oracle (<http://www.oracle.com/technetwork/java>)



1º Laboratório ECOP15

31-08 e 06-09 de 2021

Exemplo1 – Entrada de dados pelo console

```
// Addition program that displays the sum of two numbers.
import java.util.Scanner; // program uses class Scanner
public class Addition
{
    // main method begins execution of Java application
    public static void main( String args[] )
    { // create Scanner to obtain input from command window
        Scanner input = new Scanner( System.in );
        int number1; // first number to add
        int number2; // second number to add
        int sum; // sum of number1 and number2
        System.out.print( "Enter first integer: " ); // prompt
        number1 = input.nextInt(); // read first number from user

        System.out.print( "Enter second integer: " ); // prompt
        number2 = input.nextInt(); // read second number from user
        sum = number1 + number2; // add numbers
        System.out.printf( "Sum is %d\n", sum ); // display sum
    } // end method main
} // end class Addition
```

Entrada e saída em console.

```
1 // Figura 2.7: Addition.java
2 // Programa de adição que insere dois números, então exibe a soma deles.
3 import java.util.Scanner; // programa utiliza a classe Scanner
4
5 public class Addition
6 {
7     // método main inicia a execução do aplicativo Java
8     public static void main(String[] args)
9     {
10         // cria um Scanner para obter entrada a partir da janela de comando
11         Scanner input = new Scanner(System.in);
12
13         int number1; // primeiro número a somar
14         int number2; // segundo número a somar
15         int sum; // soma de number1 e number2
16
17         System.out.print("Enter first integer: "); // prompt
18         number1 = input.nextInt(); // lê primeiro o número fornecido pelo usuário
19
20         System.out.print("Enter second integer: "); // prompt
21         number2 = input.nextInt(); // lê o segundo número fornecido pelo usuário
22
23         sum = number1 + number2; // soma os números, depois armazena o total em sum
24
25         System.out.printf("Sum is %d\n", sum); // exibe a soma
26     } // fim do método main
27 } // fim da classe Addition
```

Entrada e saída em Diálogo

```
1 // Figura 12.2: Addition.java
2 // Programa de adição que utiliza JOptionPane para entrada e saída.
3 import javax.swing.JOptionPane;
4
5 public class Addition
6 {
7     public static void main(String[] args)
8     {
9         // obtém a entrada de usuário a partir dos diálogos de entrada JOptionPane
10        String firstNumber =
11            JOptionPane.showInputDialog("Enter first integer");
12        String secondNumber =
13            JOptionPane.showInputDialog("Enter second integer");
14
15        // converte String em valores int para utilização em um cálculo
16        int number1 = Integer.parseInt(firstNumber);
17        int number2 = Integer.parseInt(secondNumber);
18
19        int sum = number1 + number2;
20
21        // exibe o resultado em um diálogo de mensagem JOptionPane
22        JOptionPane.showMessageDialog(null, "The sum is " + sum,
23            "Sum of Two Integers", JOptionPane.PLAIN_MESSAGE);
24    }
25 } // fim da classe Addition
```




Exercícios

1. Escreva um programa em Java que desenhe um Retângulo Na Tela, com as seguintes personalizações possíveis:
 - Possua **largura** e **altura** lidas pelo teclado;
 - Leia o caractere que vai ser usado para desenhar a **borda**;
 - Leia o caractere de **preenchimento**;
 - Pergunte ao usuário se o retângulo vai ser **preenchido** ou não.

Como exemplo, se o programa executasse com largura = 12, altura = 5, com o caractere **x** para borda, o caractere **.** para preenchimento e o retângulo fosse preenchido, o resultado seria o mostrado abaixo:

```
xxxxxxxxxxxxxxxx
x.....x
x.....x
x.....x
x.....x
xxxxxxxxxxxxxxxx
```



Exercícios

2. Crie um programa em Java calcular e imprimir os 20 primeiros números primos. Lembrar que um número primo é aquele que só é divisível por ele mesmo e por um.

3. Faça um aplicativo Java para calcular o fatorial de um número, digitado pelo usuário. Lembrar que :

$$n! = 1 * 2 * \dots * (n-2) * (n-1) * n;$$



Exercícios

4. Crie um método em Java que calcule a distância ***D*** (em metros) alcançada por um míssil lançado com uma velocidade inicial ***V*** (em m/s) e um ângulo de tiro ***A*** (em graus). Sabe-se da física básica que a expressão para esse cálculo (desconsiderando o atrito do ar) é a seguinte:

$$D = \frac{V^2 \cdot \sin(2A)}{g} \quad \text{onde } g = 9,8 \text{ m/s}^2$$



Exercícios

- 5.** (Imprimindo o Equivalente Decimal de um Número Binário)
Escreva um aplicativo que insira um **inteiro** contendo apenas 0s e 1s (isto é, um inteiro binário) e imprima seu equivalente decimal. [Dica: use os operadores de resto e de divisão para selecionar os dígitos do número binário, um de cada vez, da direita para a esquerda. No sistema numérico decimal, o dígito mais à direita tem um valor posicional de 1 e o próximo dígito à esquerda um valor posicional de 10, depois 100, depois 1000, e assim por diante. O número decimal 234 pode ser interpretado como $4 * 1 + 3 * 10 + 2 * 100$. No sistema numérico binário, o dígito mais à direita tem um valor posicional de 1, o próximo dígito à esquerda um valor posicional de 2, então 4, depois 8 e assim por diante. O equivalente decimal do binário 1101 é $1 * 1 + 0 * 2 + 1 * 4 + 1 * 8$, ou $1 + 0 + 4 + 8$ ou, 13.]



Exercícios

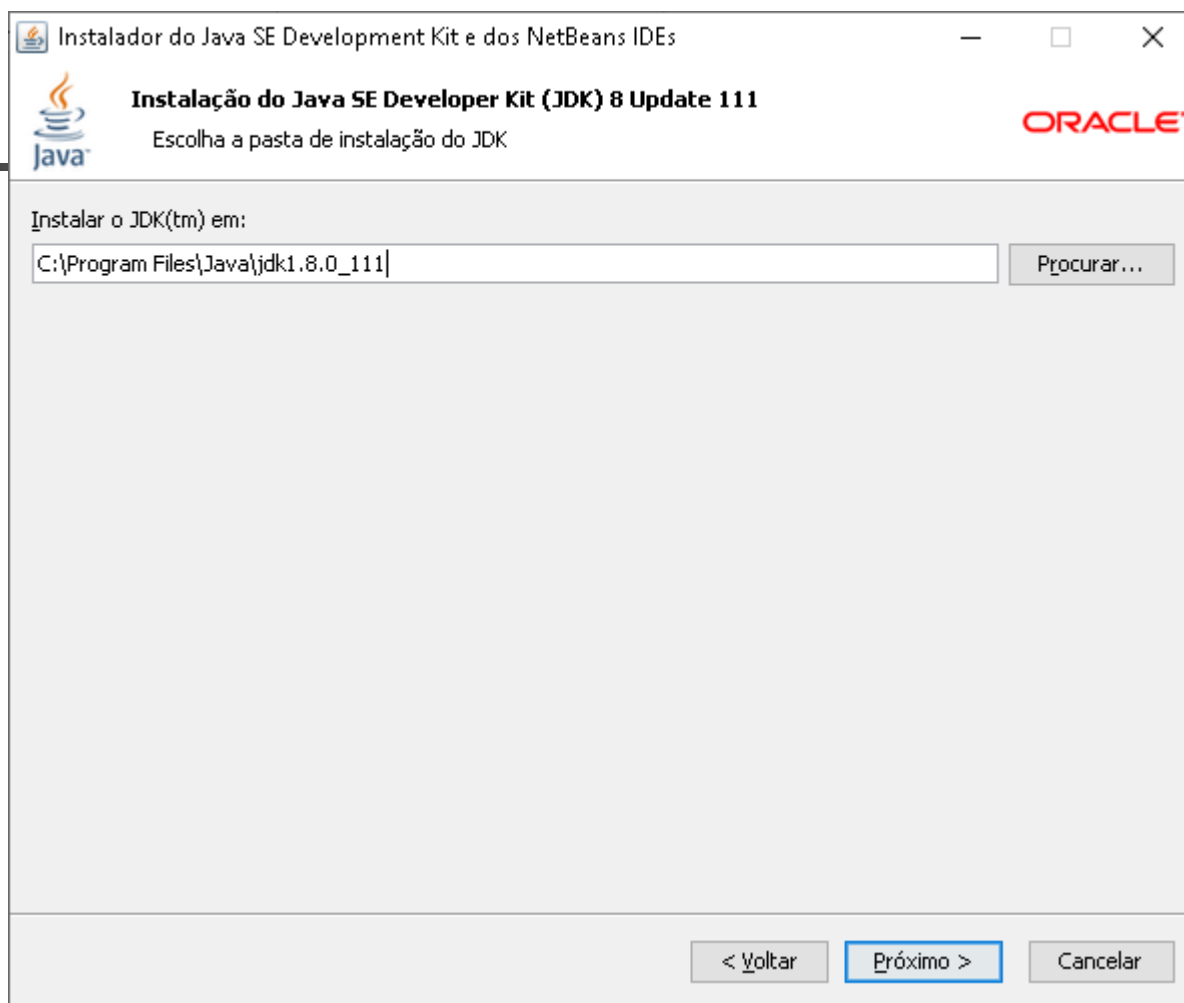
6. (Palíndromo) Um palíndromo é uma sequência de caracteres na qual é possível ler-se o mesmo conteúdo, tanto do início até o final como do final até o início. Por exemplo, cada um dos cinco números inteiros seguintes é um palíndromo: 12321, 55555, 45554 e 11611. Escreva um aplicativo que leia um inteiro de cinco dígitos e determine se é um palíndromo. Se o número não tiver cinco dígitos, exiba uma mensagem de erro e permita que o usuário insira um novo valor.

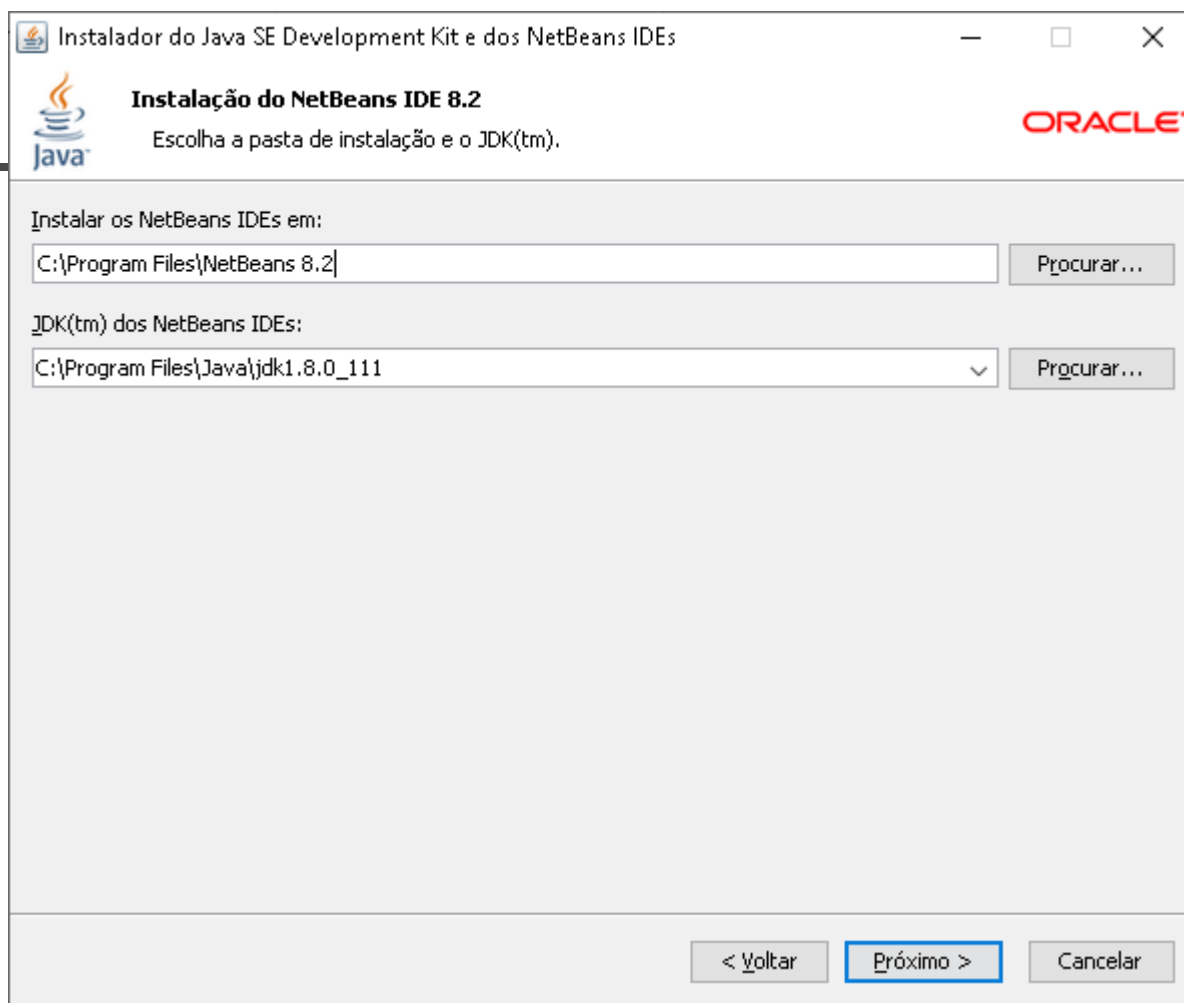


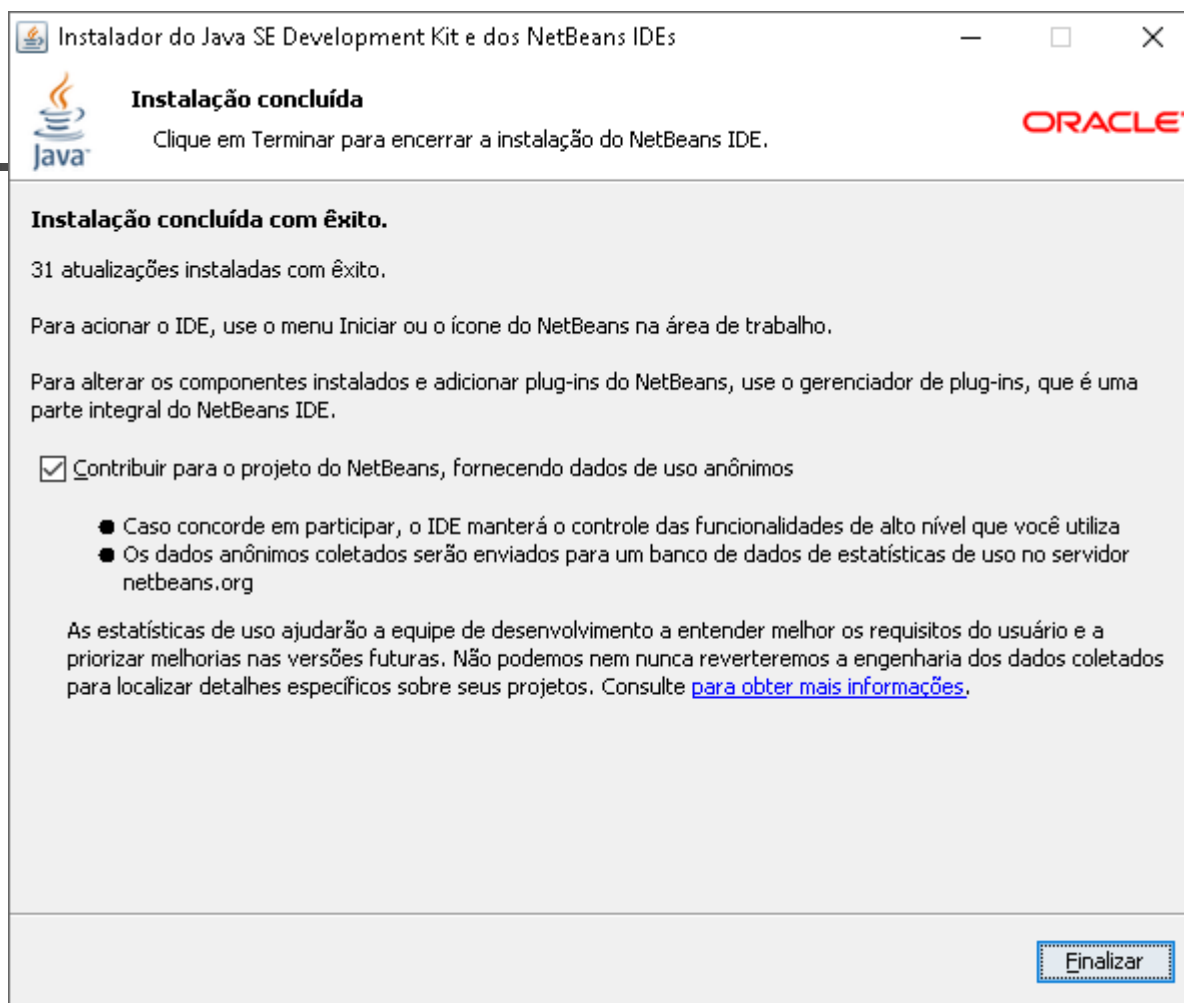
Instalando o NetBeans 8.2

- <https://www.oracle.com/technetwork/java/javase/downloads/jdk-netbeans-jsp-3413139-esa.html>











Criando um projeto



Aprender & Descobrir

Meu NetBeans

Novidades

Mostrar na Inicialização ☒

Aprender & Descobrir

[Faça um Tour](#)[Experimente um Projeto de Amostra](#)[Novidades](#)[Local da Comunidade](#)

Demonstrações & Tutoriais

[Aplicações do Java SE](#)[Aplicações da GUI Java e JavaFX](#)[Aplicações Java EE e Java Web](#)[Aplicações C/C++](#)[Aplicações PHP e HTML5](#)[Aplicações Móveis e Incorporadas](#)[Toda a Documentação On-line >>](#)

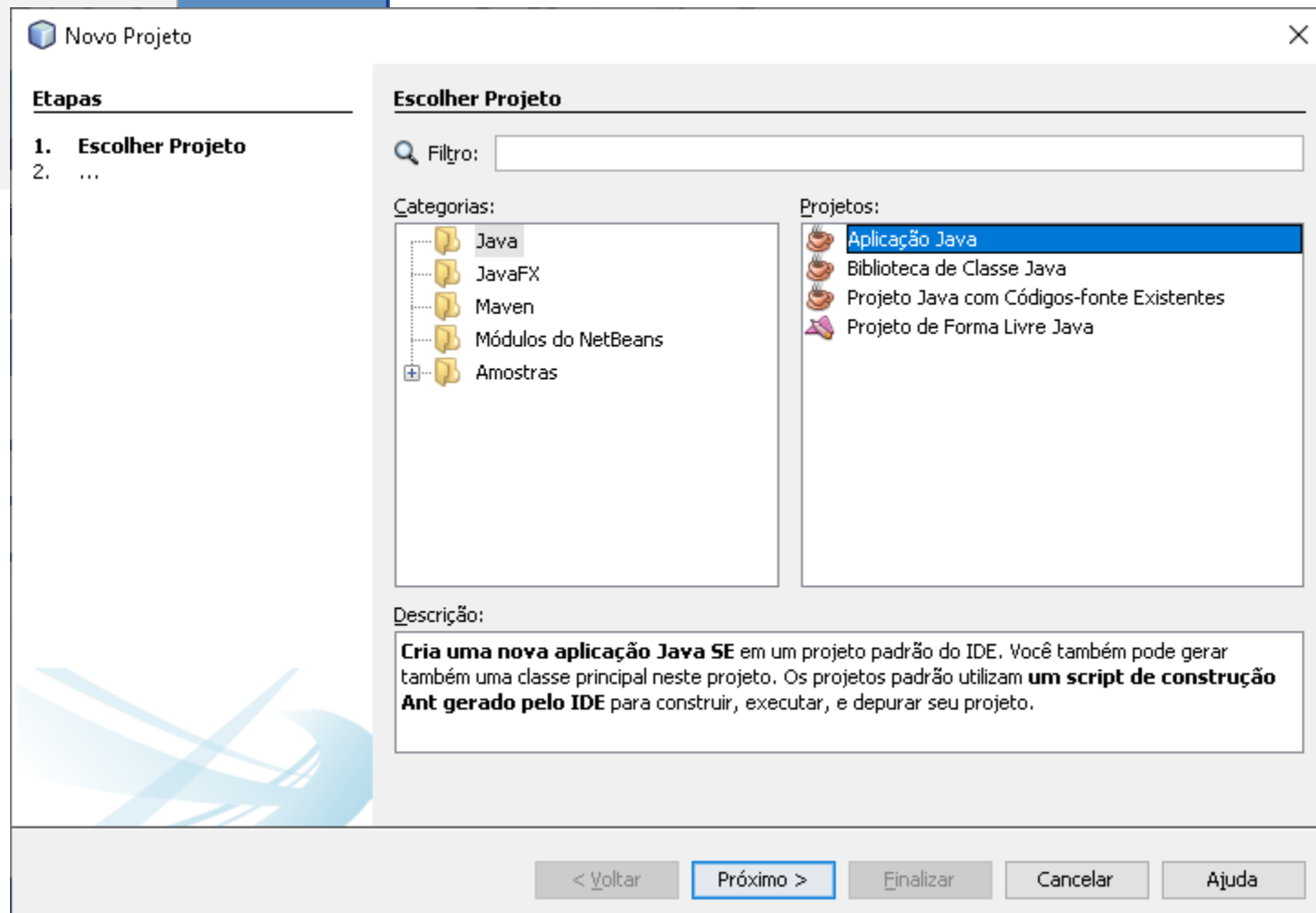
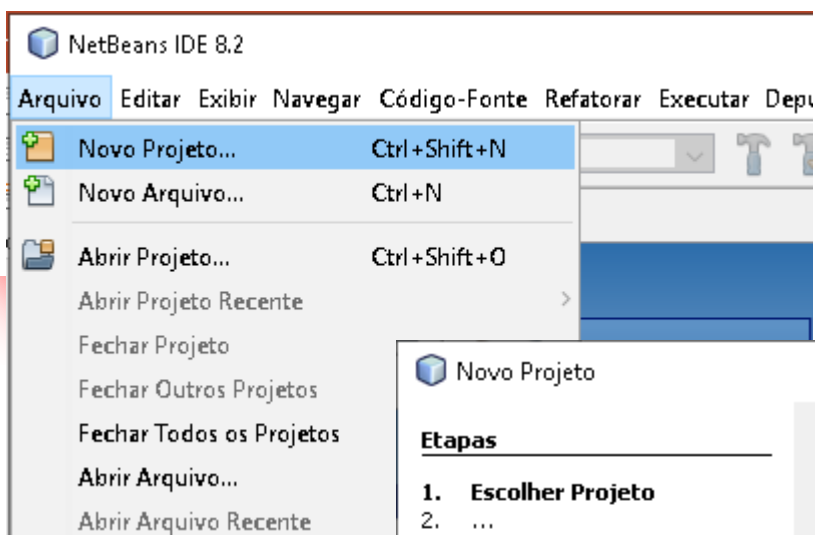
Demonstração em Destaque



Hello Apache NetBeans 11.0!

ORACLE





Novo Aplicação Java

Etapas

1. Escolher Projeto
2. **Nome e Localização**

Nome e Localização

Nome do Projeto:

Localização do Projeto:

Procurar...

Pasta do Projeto:

☐ Usar Pasta Dedicada para Armazenar Bibliotecas

Pasta Bibliotecas:

Procurar...

Usuários e projetos diferentes podem compartilhar as mesmas bibliotecas de compilação (consulte a Ajuda para obter detalhes).

☒ Criar Classe Principal

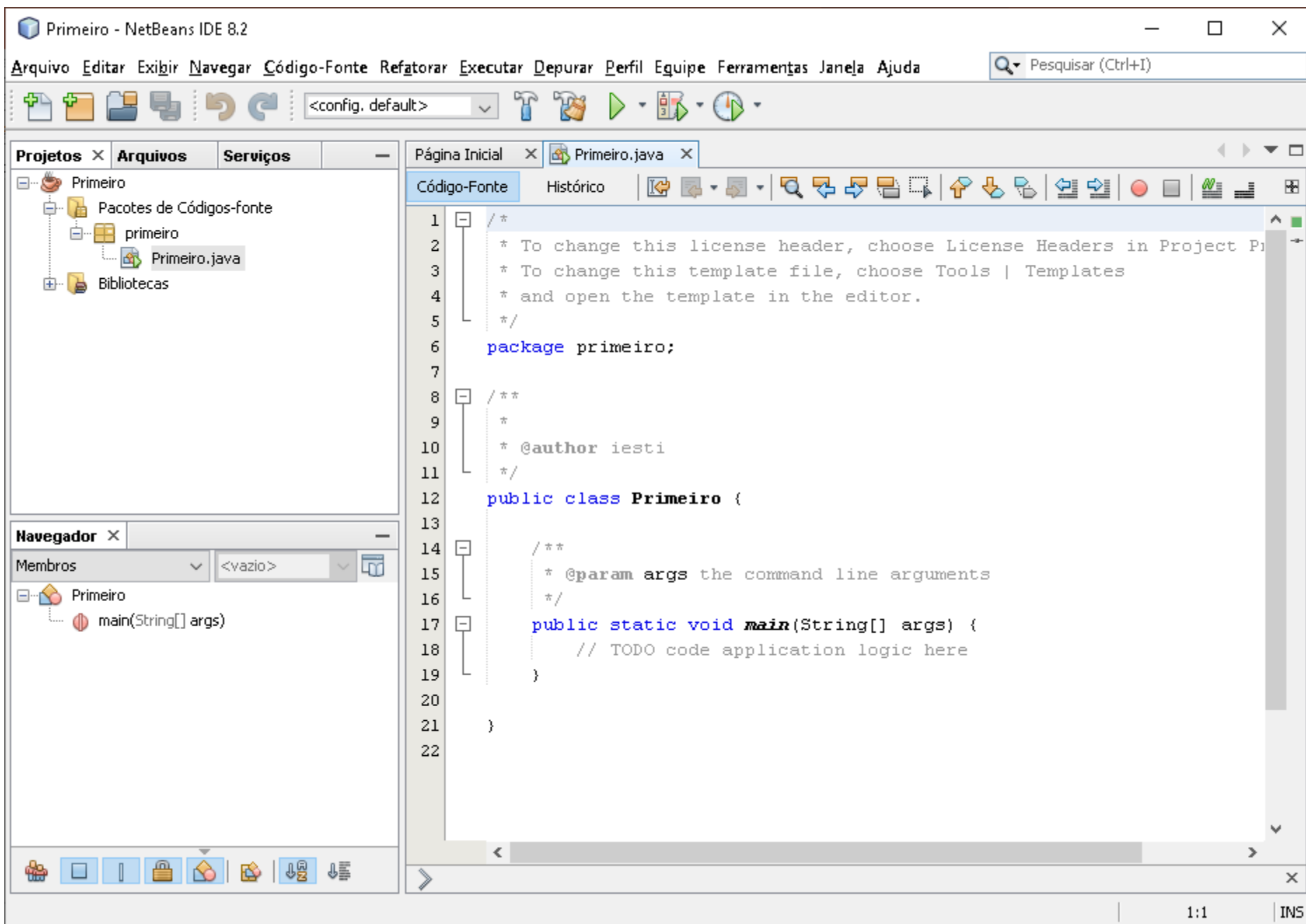
< Voltar

Próximo >

Finalizar

Cancelar

Ajuda





Configurar para uso no prompt



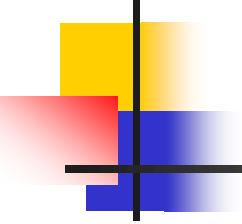
Verificando a instalação

```
Prompt de Comando
Microsoft Windows [versão 10.0.18362.1016]
(c) 2019 Microsoft Corporation. Todos os direitos reservados.

C:\Users\iesti>java -version
java version "1.8.0_251"
Java(TM) SE Runtime Environment (build 1.8.0_251-b08)
Java HotSpot(TM) 64-Bit Server VM (build 25.251-b08, mixed mode)

C:\Users\iesti>javac
'javac' não é reconhecido como um comando interno
ou externo, um programa operável ou um arquivo em lotes.

C:\Users\iesti>_
```

- 
-
- Configurar as variáveis de ambiente:
 - PATH
 - JAVA_HOME
 - Colocar o diretório bin de instalação da versão do JDK do java.

Painel de Controle\Sistema e Segurança\Sistema

← → ↕ ↗ > Painel de Controle > Sistema e Segurança > Sistema

Arquivo Editar Exibir Ferramentas

EPSON Easy Photo Print Photo Print

Início do Painel de Controle

- Gerenciador de Dispositivos
- Configurações remotas
- Proteção do sistema
- Configurações avançadas do sistema**

Exibir informações básicas sobre o computador

Edição do Windows

Windows 10 Home Single Language

© 2019 Microsoft Corporation. Todos os direitos reservados.

Windows 10

Sistema

Processador: Intel(R) Core(TM) i3-4030U CPU @ 1.90GHz 1.90 GHz

Memória instalada (RAM): 4,00 GB (utilizável: 3,88 GB)

Tipo de sistema: Sistema Operacional de 64 bits, processador com base em x64

Caneta e Toque: Nenhuma Entrada à Caneta ou por Toque está disponível para este vídeo

Nome do computador, domínio e configurações de grupo de trabalho

Nome do computador: DESKTOP-6P325UK

Nome completo do computador: DESKTOP-6P325UK

Descrição do computador:

Grupo de trabalho: WORKGROUP

Ativação do Windows

Windows ativado [Ler os Termos de Licença para Software Microsoft](#)

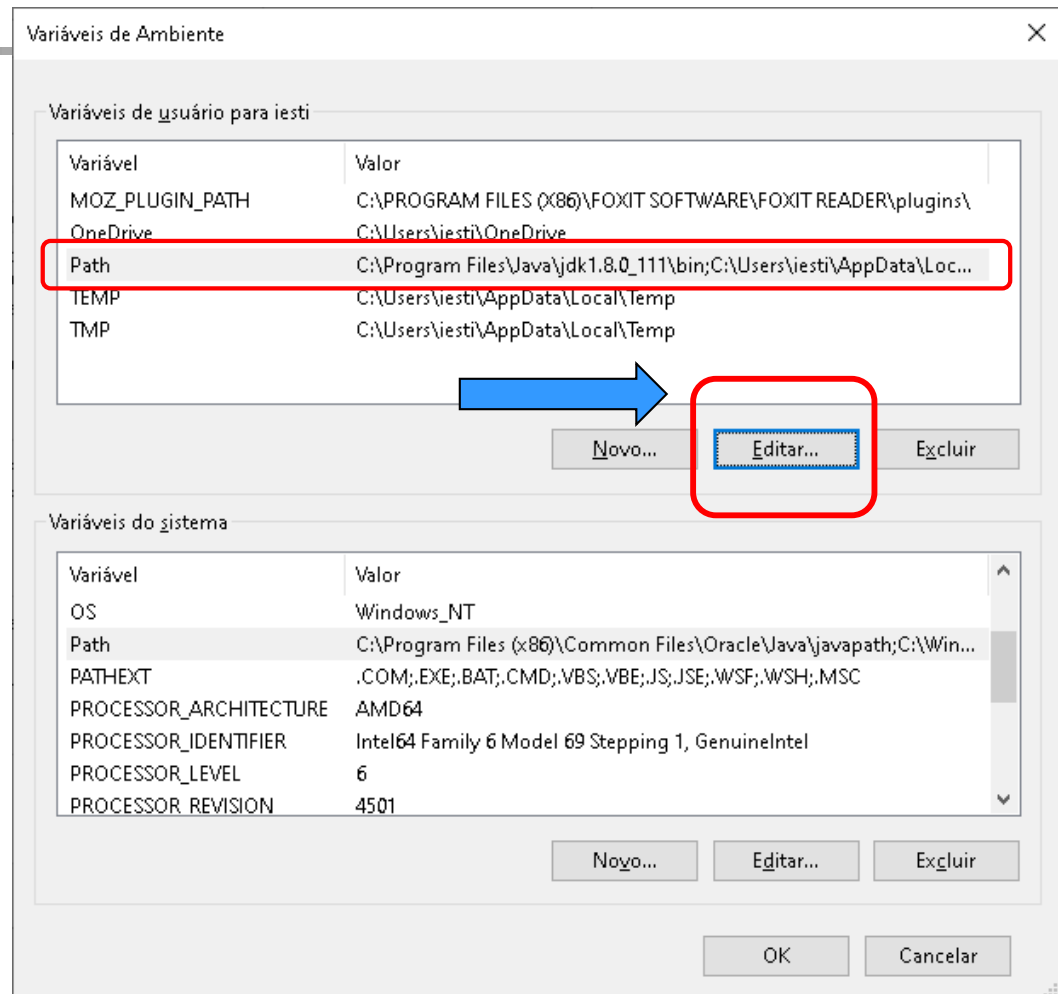
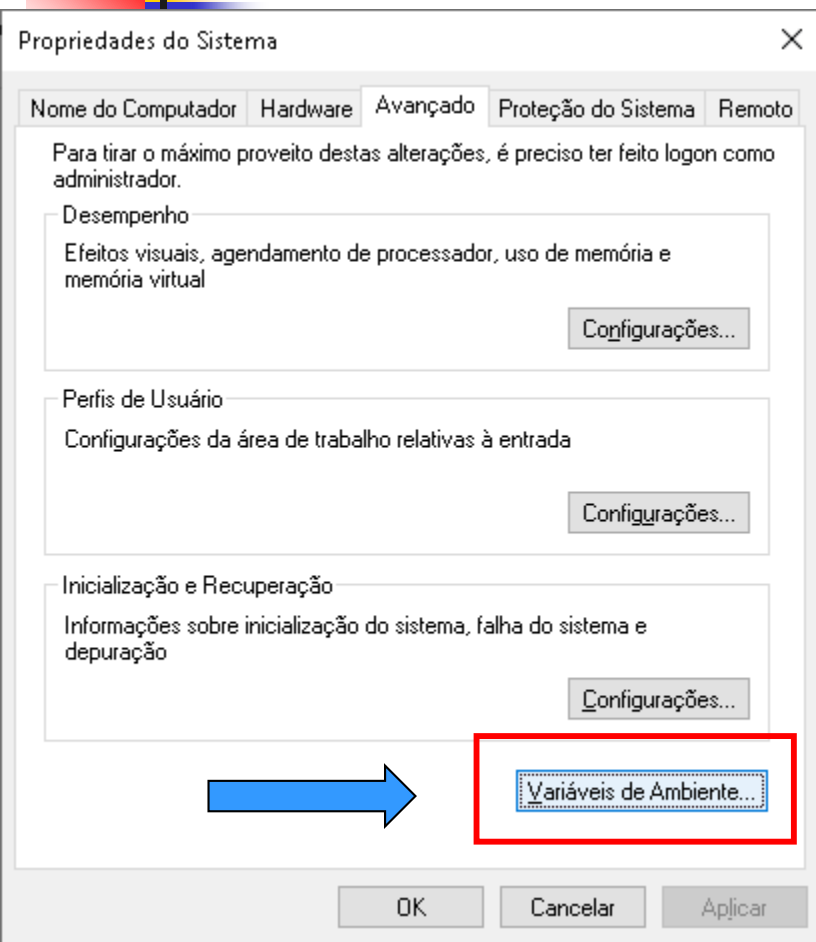
ID do Produto (Product ID): 00327-30432-24982-AAOEM

Consulte também

Segurança e Manutenção

[Alterar configurações](#)

[Alterar chave do produto \(Product Key\)](#)




```
CA Prompt de Comando
Microsoft Windows [versão 10.0.18362.1016]
(c) 2019 Microsoft Corporation. Todos os direitos reservados.

C:\Users\iesti>javac
Usage: javac <options> <source files>
Where possible options include:
  -g                      Generate all debugging info
  -g:none                 Generate no debugging info
  -g:{lines,vars,source}  Generate only some debugging info
  -nowarn                 Generate no warnings
  -verbose                Output messages about what the compiler is doing
  -deprecation            Output source locations where deprecated APIs are used
  -classpath <path>       Specify where to find user class files and annotation processors
  -cp <path>              Specify where to find user class files and annotation processors
  -sourcepath <path>       Specify where to find input source files
  -bootclasspath <path>   Override location of bootstrap class files
  -extdirs <dirs>         Override location of installed extensions
  -endorseddirs <dirs>    Override location of endorsed standards path
  -proc:{none,only}       Control whether annotation processing and/or compilation is done.
  -processor <class1>[,<class2>,<class3>...] Names of the annotation processors to run; bypasses default discovery process
  -processorpath <path>   Specify where to find annotation processors
  -parameters             Generate metadata for reflection on method parameters
  -d <directory>          Specify where to place generated class files
  -s <directory>          Specify where to place generated source files
  -h <directory>          Specify where to place generated native header files
  -implicit:{none,class}  Specify whether or not to generate class files for implicitly referenced files
  -encoding <encoding>    Specify character encoding used by source files
  -source <release>        Provide source compatibility with specified release
  -target <release>        Generate class files for specific VM version
```