

APRESENTANDO A LINGUAGEM DE PROGRAMAÇÃO JAVA

Um guia rápido sobre a linguagem de
programação

O QUE É JAVA

História e principais características

A história da linguagem Java

- Criada em 1995, um projeto da Sun chamado de Green, para ser uma linguagem voltada a sistemas embarcados em produtos eletrônicos como televisores
 - Nome dos criadores: James Gosling, Patrick Naughton, Mike Sheridan
- Seu primeiro nome foi Oak (carvalho) devido a uma árvore que ficava em frente a janela do criador da linguagem
- Porém dois problemas:
 1. Já existia uma linguagem com o nome Oak
 2. O projeto para produtos eletrônicos não vingou



A história da linguagem Java



- O caminho parecia ser a descontinuidade da ideia, mas enxergando a potencialidade da linguagem, a Sun continuou a investir ampliando o escopo de sua atuação
- A linguagem recebeu então o nome de Java, em homenagem ao café consumido pelos desenvolvedores
 - Uma lista de nomes foi criada e averiguada por advogados. Java era o 4º da lista
- Dado o boom da Internet em 1993 e a necessidade em criar páginas que não fossem mais estáticas, o Java ganhou espaço e força no desenvolvimento de conteúdo dinâmico para websites (primórdios da Web 2.0)
 - Não confundir Java com Javascript!

A história da linguagem Java

- Java conta com uma série de ferramentas/tecnologias para desenvolvimento
- Em sua criação já contava com:
 - Applets
 - AWT e Swing
 - Servlets
 - JDBC
- Novas tecnologias foram incluídas com o passar dos anos como:
 - EJB
 - Implementação de Web Services
 - JSF
 - JavaFX
- Desde o início já utilizada o conceito de máquina virtual (veremos adiante com mais detalhes)

Pacotes de desenvolvimento



- Para desenvolver em Java, o programador deve antes escolher para qual ambiente (escopo) deseja codificar. Ao todo são três kits para desenvolvimento disponíveis para o download:
 - Java SE – Java Standard Edition
 - Java EE – Java Enterprise Edition
 - Java ME – Java Mobile Edition

The screenshot shows the Oracle Java SE Downloads page. The Oracle logo is in the top left. Navigation links include Sign In/Register, Help, Country, Communities, I am a..., I want to..., and Search. Below these are links for Products, Solutions, Downloads, Store, Support, Training, and Partners. The breadcrumb trail reads: Oracle Technology Network > Java > Java SE > Downloads. A sidebar on the left lists various Java-related links: Java SE, Java EE, Java ME, Java SE Support, Java SE Advanced & Suite, Java Embedded, Java DB, Web Tier, Java Card, Java TV, New to Java, Community, and Java Magazine. The main content area is titled 'Java SE Downloads' and features two download buttons: 'DOWNLOAD' for Java Platform (JDK) 8u40 and 'DOWNLOAD' for NetBeans with JDK 8. Below this, a section titled 'Java Platform, Standard Edition' provides details about the Java SE 8u40 release, including its benefits and a link to learn more. On the right side of this section, there are three download buttons: 'JDK DOWNLOAD', 'Server JRE DOWNLOAD', and 'JRE DOWNLOAD'.



Linguagens compiladas e interpretadas

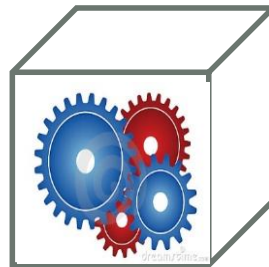
- Linguagens Compiladas: o código fonte é traduzido para uma linguagem de máquina e executada diretamente pelo Sistema Operacional.
 - Primeiro o código fonte é “traduzido” totalmente para um executável
 - Depois é executado pelo Sistema Operacional
 - É necessário um compilador para cada Sistema Operacional que se pretende rodar
 - Não se tem acesso ao código fonte na versão compilada da aplicação
 - A execução da aplicação é mais rápida
- Exemplo de linguagens compiladas: C, C++, Pascal, Fortran, Delphi

```
#include <stdio.h>
#include <windows.h>

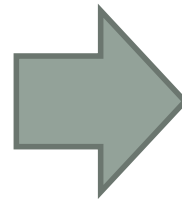
int main(int argc, char *argv[])
{
    if(argc != 3)
    {
        printf("use atqwin <alvo> <comando>");
        return 1;
    }

    atqwin(argc, argv);
}
```

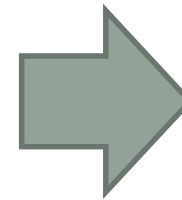
Código Fonte



Compilador



Código objeto



Código executável

Linguagens compiladas e interpretadas

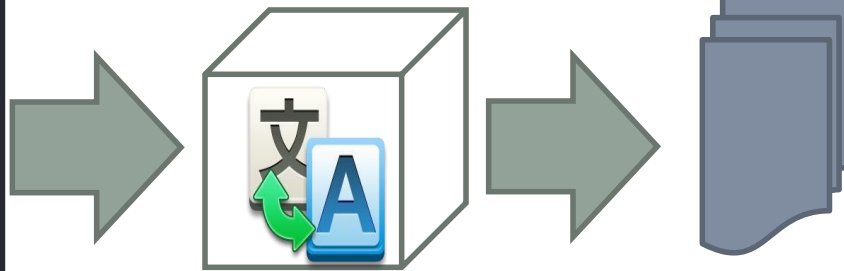
- Linguagens interpretadas: o código fonte é executado (interpretado) diretamente sem passar para uma linguagem de máquina. Utiliza o conceito de máquina virtual (ou algum aplicativo intermediário) que executa os comandos de alto nível e repassa ao Sistema Operacional em execução
 - Possui performance mais lenta que as linguagens compiladas
 - Possui maior facilidade de portabilidade, devido o uso da máquina virtual / aplicativo intermediário
 - É possível realizar ajustes / depuração com o programa em execução
 - Não se tem acesso ao código fonte na versão compilada da aplicação
- Exemplo de linguagens interpretadas: C#, PHP, Javascript, Python

```
#include <stdio.h>
#include <windows.h>

int main(int argc, char *argv[])
{
    if(argc != 3)
    {
        printf("use atqwin <alvo> <comando>");
        return 1;
    }

    atqwin(argc, argv);
}
```

Código Fonte

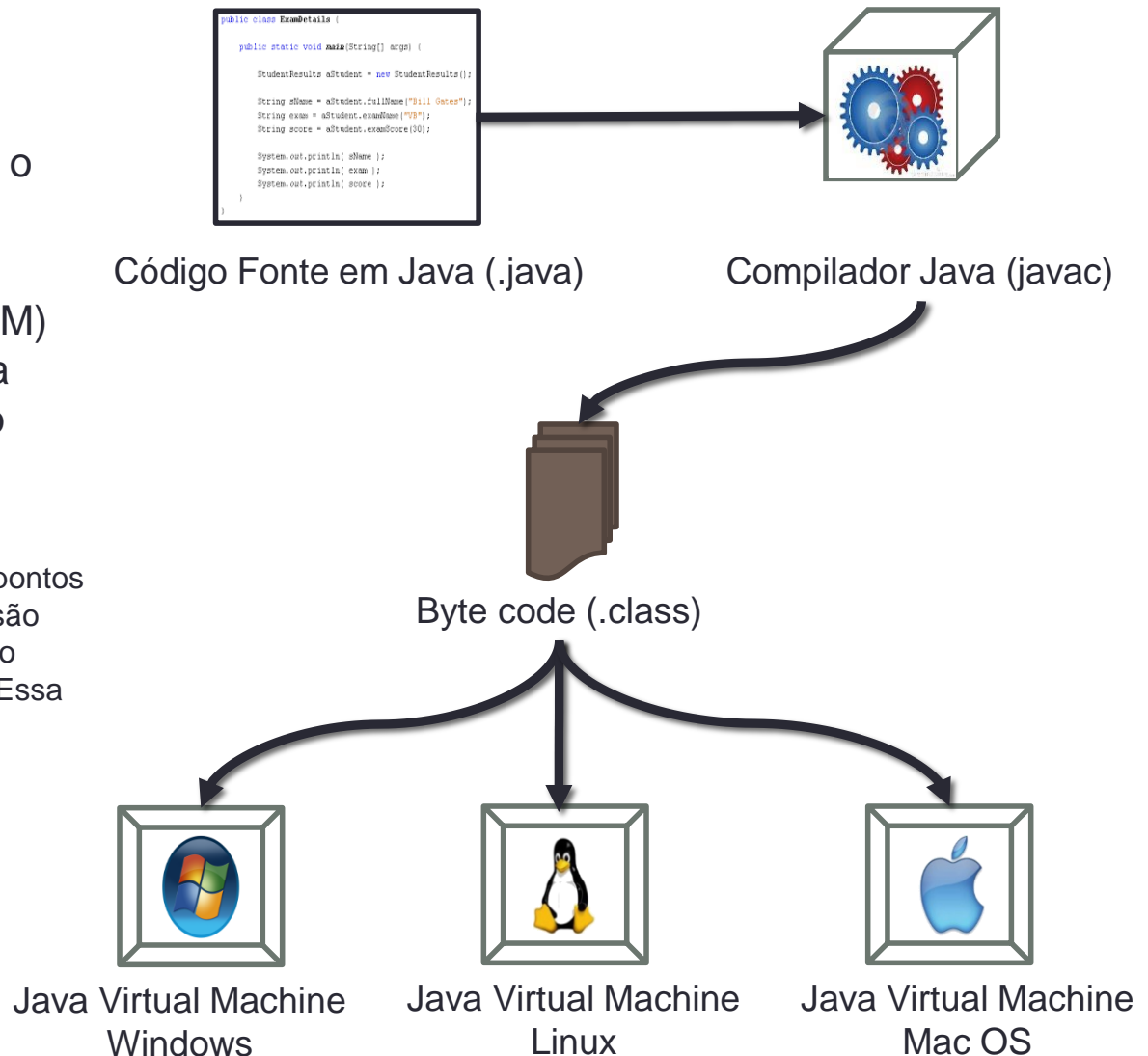


Interpretador
(intermediário)

Trecho de programa
executável (output)

Linguagens compiladas e interpretadas

- Java é compilada e interpretada
- O código é compilado para o Bytecode e interpretado na Máquina Virtual
- A Máquina Virtual Java (JVM) possui implementação para otimização da execução do código
- Estratégia adaptativa
 - Trechos chamados hot spots – pontos do código de maior atividade – são identificados e esses trechos são compilados para código nativo. Essa técnica se chama Just in Time Compiling.



COMO ESCREVER PROGRAMAS EM JAVA

Configurações e instalações de IDEs

A JDK – Java Development Kit

- É necessário fazer o download do Java Development Kit apropriado para começar a escrever os seus programas
- Em nosso curso usaremos a versão Java SE – Standard Edition
- Download em:
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Além do JDK, é aconselhável a utilização de uma ferramenta de ambiente para programação: IDE

IDE para Java (Integrated Development Environment)

- Existem diversas IDEs para desenvolvimento em Java. Abaixo segue algumas delas e seus links para download



<http://netbeans.org/downloads/>



<http://eclipse.org/downloads/>



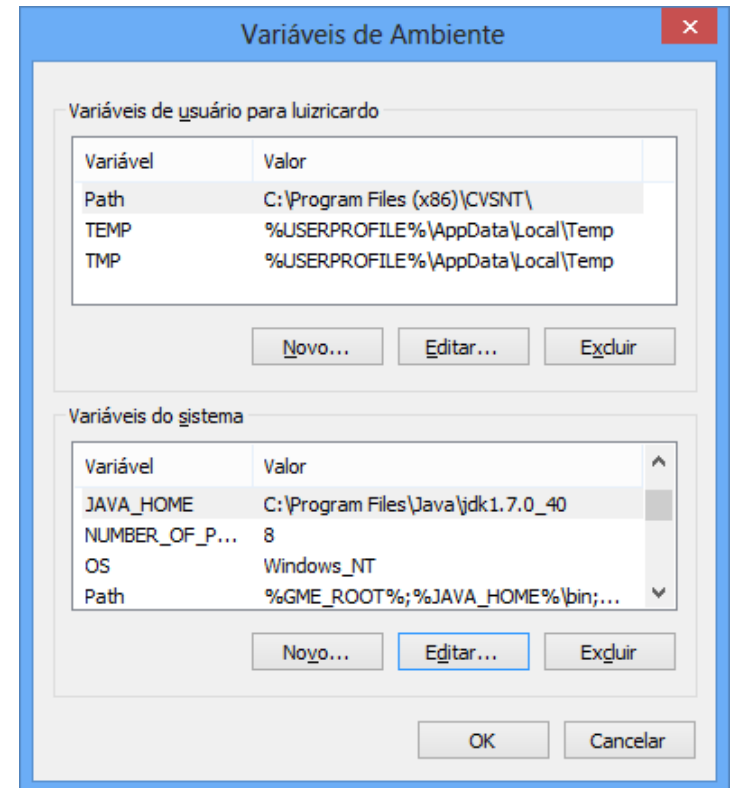
<http://www.bluej.org/>

Como compilar seus códigos: via prompt

- Primeiro modo de compilar um programa em Java é via prompt de comando
- Após instalar o JDK é possível acessar ao conjunto de executáveis do Java
- Para acessar de qualquer diretório, certifique-se de que a **variável de ambiente está configurada** para o diretório do Java
 - Senão, será necessário executar diretamente do diretório do Java
- O código fonte será escrito no Notepad como qualquer texto e salvo com a extensão .java
 - Se salvar como .txt o compilador Java não reconhecerá o arquivo!

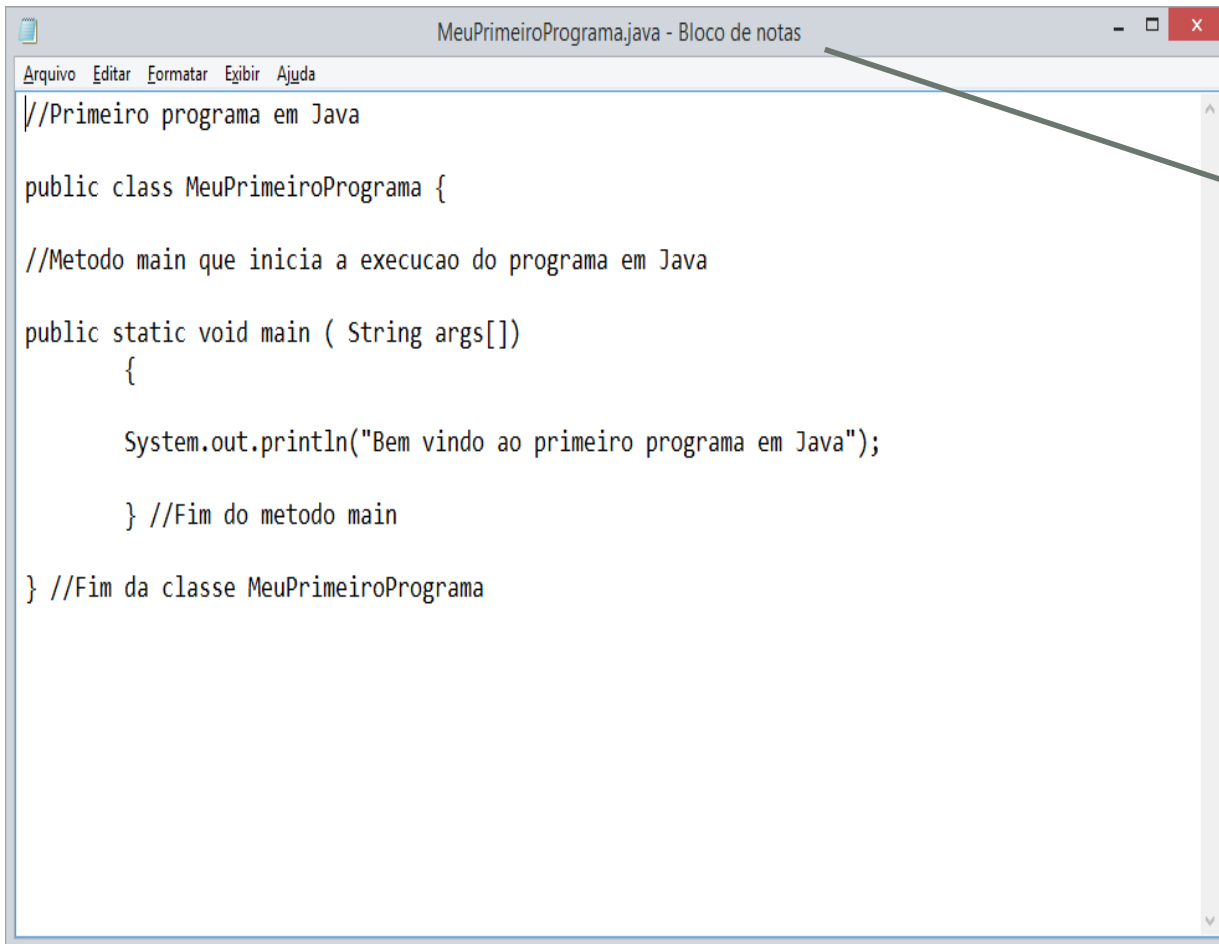
Como compilar seus códigos: via prompt

- Como configurar a variável de ambiente no Windows:
- Windows 7 e 8:
https://youtu.be/Rz6KZr_O62U
- Windows 10:
<https://youtu.be/dhDCoJZVw1I>



Como compilar seus códigos: via prompt

- Exemplo do primeiro código em Java



```
MeuPrimeiroPrograma.java - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda
//Primeiro programa em Java

public class MeuPrimeiroPrograma {

    //Metodo main que inicia a execucao do programa em Java

    public static void main ( String args[])
    {

        System.out.println("Bem vindo ao primeiro programa em Java");

    } //Fim do metodo main

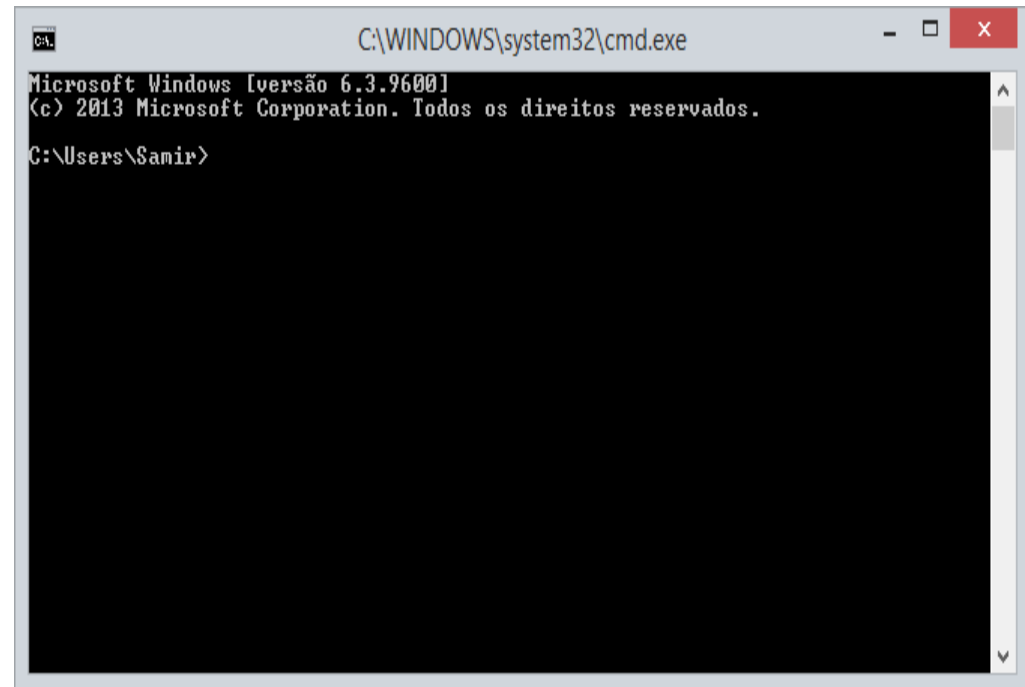
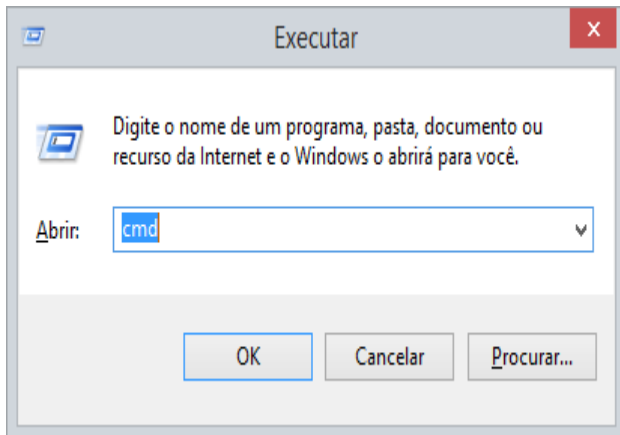
} //Fim da classe MeuPrimeiroPrograma
```

Nesse caso foi utilizado a aplicação Bloco de Notas para a escrita do código.

Após finalizar a codificação, o arquivo deve ser salvo com o nome da classe, nesse caso “MeuPrimeiroPrograma” seguido de .java (não é .txt)

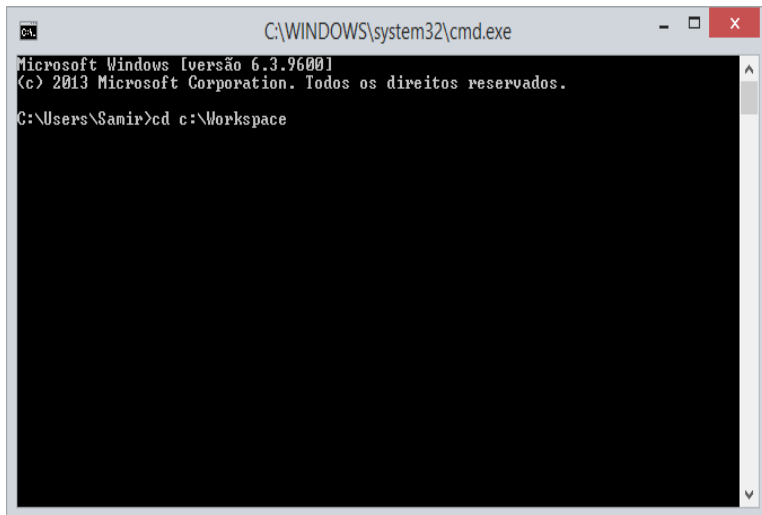
Como compilar seus códigos: via prompt

- Para compilar o primeiro programa, acesse o prompt de comando
 - Clique no botão Iniciar e digite cmd ou
 - Aperte simultaneamente as teclas Windows +R e digite: cmd

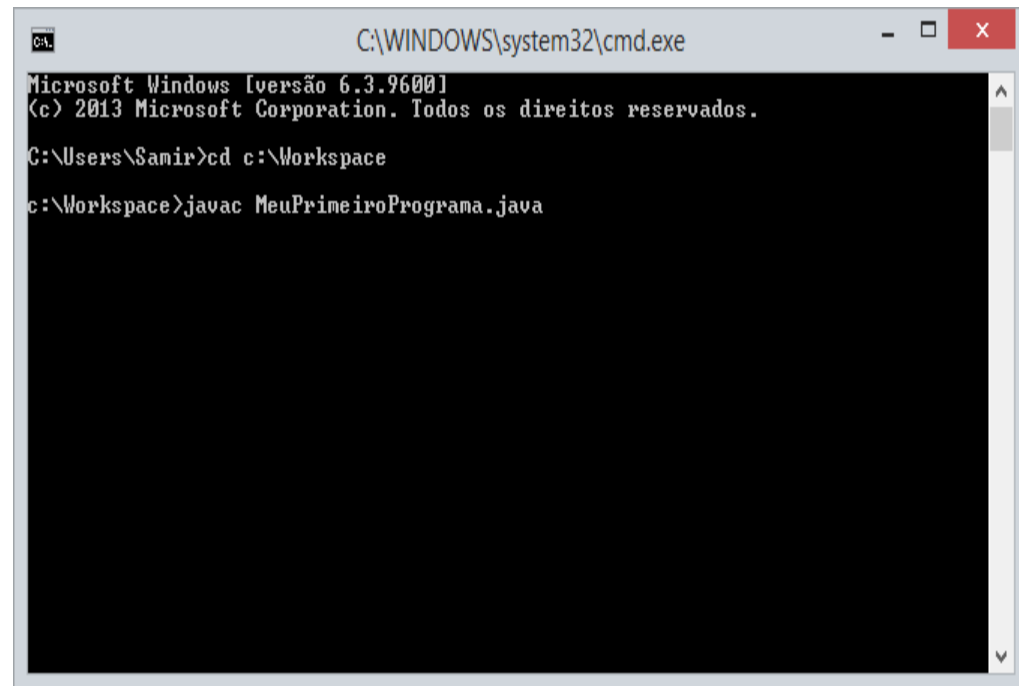


Como compilar seus códigos: via prompt

- No prompt, navegue até onde se encontra o seu arquivo com o código fonte.
 - No caso do exemplo seria navegar até a pasta c:\Workspace
- Em seguida execute o compilador do Java com o comando:
 - `javac MeuPrimeiroPrograma.java`



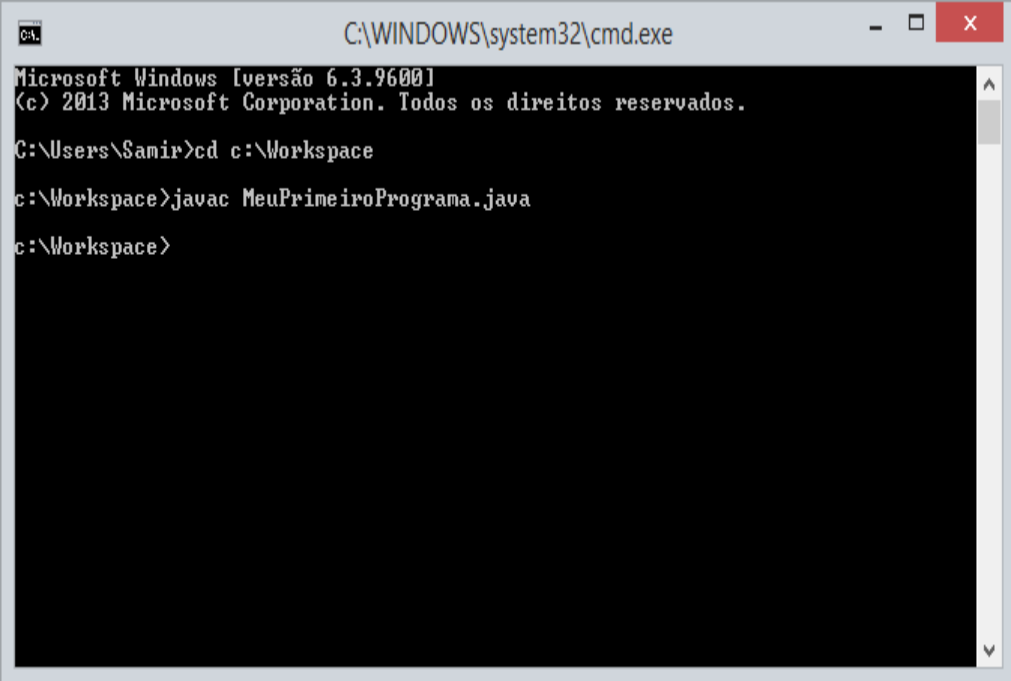
```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [versão 6.3.9600]
(c) 2013 Microsoft Corporation. Todos os direitos reservados.
C:\Users\Samir>cd c:\Workspace
```



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [versão 6.3.9600]
(c) 2013 Microsoft Corporation. Todos os direitos reservados.
C:\Users\Samir>cd c:\Workspace
c:\Workspace>javac MeuPrimeiroPrograma.java
```

Como compilar seus códigos: via prompt

- Se o código não possuir erros, o javac executará sem mensagem alguma. Após a execução será retornada a próxima linha do prompt
- No diretório Workspace, um novo arquivo será criado: MeuPrimeiroPrograma.java



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [versão 6.3.9600]
(c) 2013 Microsoft Corporation. Todos os direitos reservados.

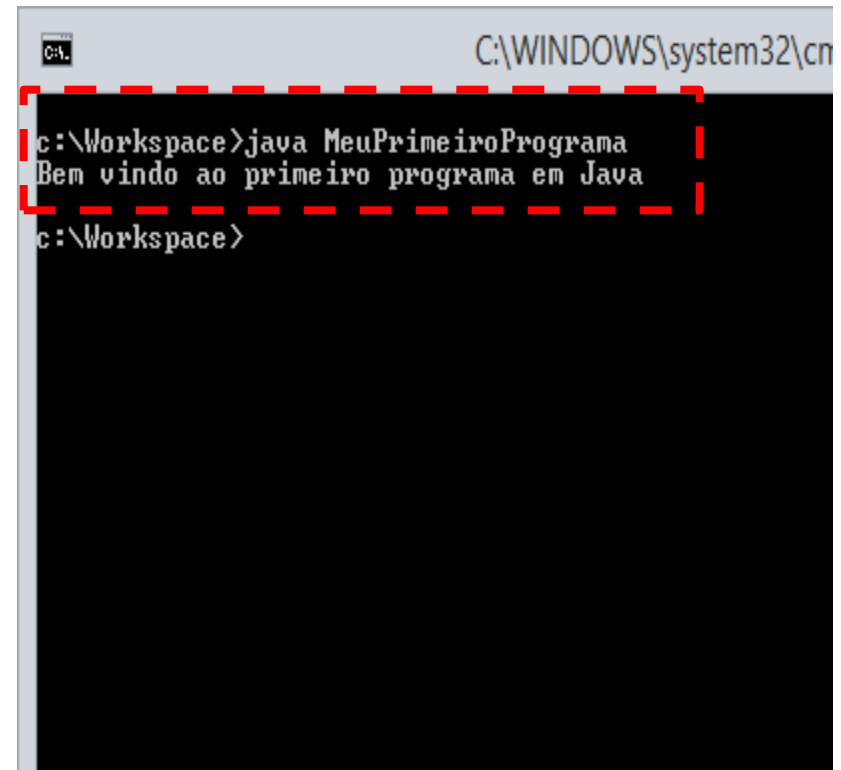
C:\Users\Samir>cd c:\Workspace

c:\Workspace>javac MeuPrimeiroPrograma.java

c:\Workspace>
```

Como compilar seus códigos: via prompt

- Para executar o programa compilado, é necessário chamar a JVM apontando para o arquivo.
- Para isso ainda no prompt digite:
java MeuPrimeiroPrograma
 - Não coloque a extensão .class para a execução
- O resultado de saída será no próprio prompt



```
C:\WINDOWS\system32\cmd.exe
c:\Workspace>java MeuPrimeiroPrograma
 Bem vindo ao primeiro programa em Java
c:\Workspace>
```

The screenshot shows a Windows command prompt window with a title bar that reads "C:\WINDOWS\system32\cmd.exe". The command prompt is at "c:\Workspace>". The user has entered the command "java MeuPrimeiroPrograma". The output of the command is " Bem vindo ao primeiro programa em Java". The command prompt is now at "c:\Workspace>". A red dashed rectangle highlights the command and its output.



prompt

WINDOWS\system32\cmd

programa
em Java

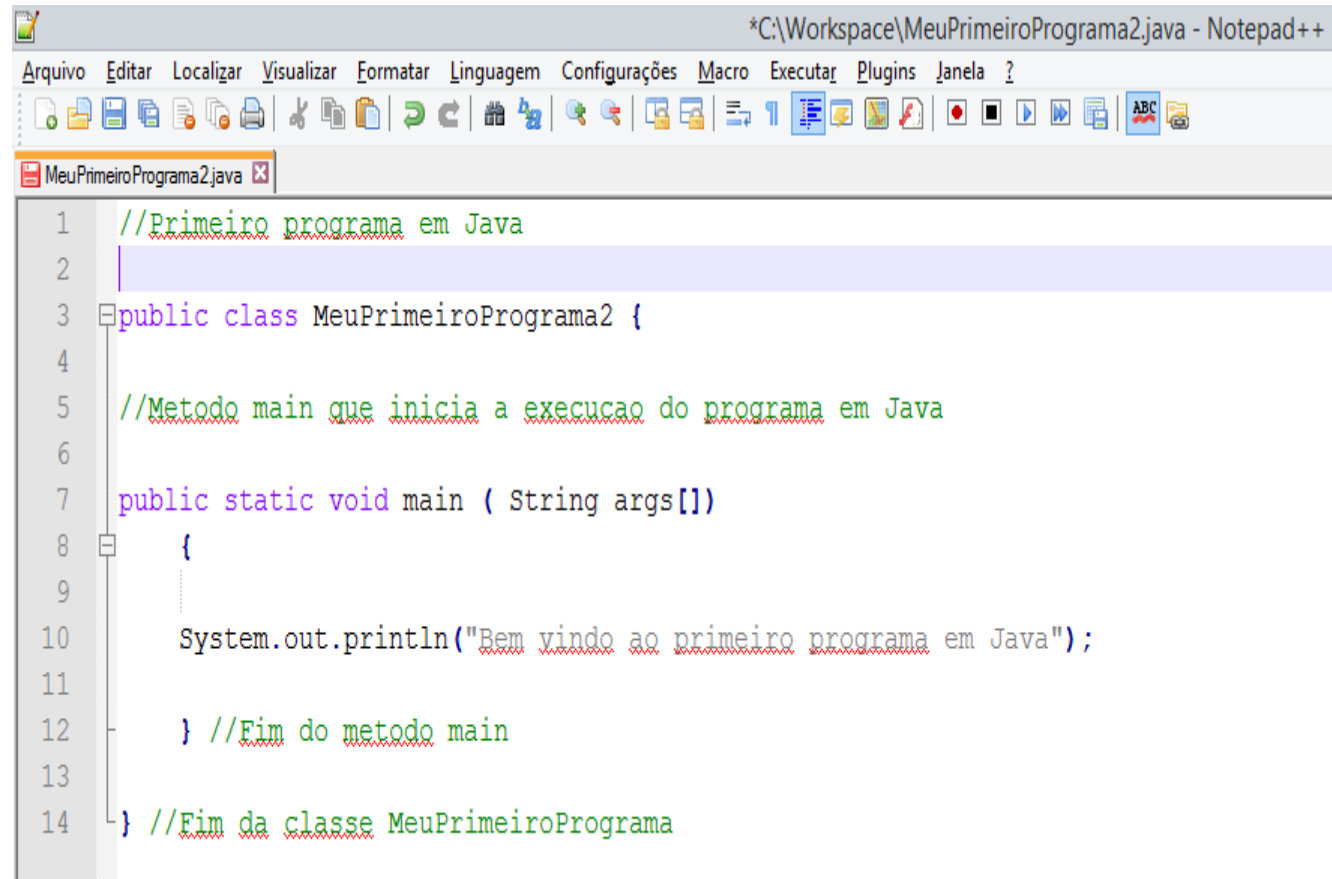
- Não coloque a extensão .class para a e

Esse eu chamo de Modo Rambo de programação

no próprio prompt

Como compilar seus códigos: via prompt

- Para facilitar um pouco a vida, pode-se utilizar editores de texto para programação.
- Esses editores não complementam código, não compilam a partir deles, apenas auxiliam colorindo os blocos de comando facilitando a leitura do código

A screenshot of the Notepad++ text editor. The title bar at the top reads '*C:\Workspace\MeuPrimeiroPrograma2.java - Notepad++'. The menu bar includes 'Arquivo', 'Editar', 'Localizar', 'Visualizar', 'Formatar', 'Linguagem', 'Configurações', 'Macro', 'Executar', 'Plugins', 'Janela', and '?'. The toolbar contains various icons for file operations, editing, and execution. The editor window shows a Java file named 'MeuPrimeiroPrograma2.java' with the following code:

```
1 //Primeiro programa em Java
2
3 public class MeuPrimeiroPrograma2 {
4
5     //Metodo main que inicia a execucao do programa em Java
6
7     public static void main ( String args[])
8     {
9         ...
10        System.out.println("Bem vindo ao primeiro programa em Java");
11
12    } //Fim do metodo main
13
14 } //Fim da classe MeuPrimeiroPrograma
```

Editores de texto inteligentes



<https://notepad-plus-plus.org/>

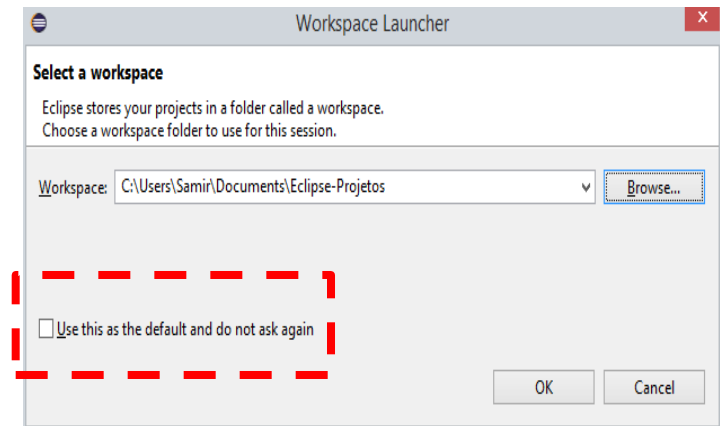
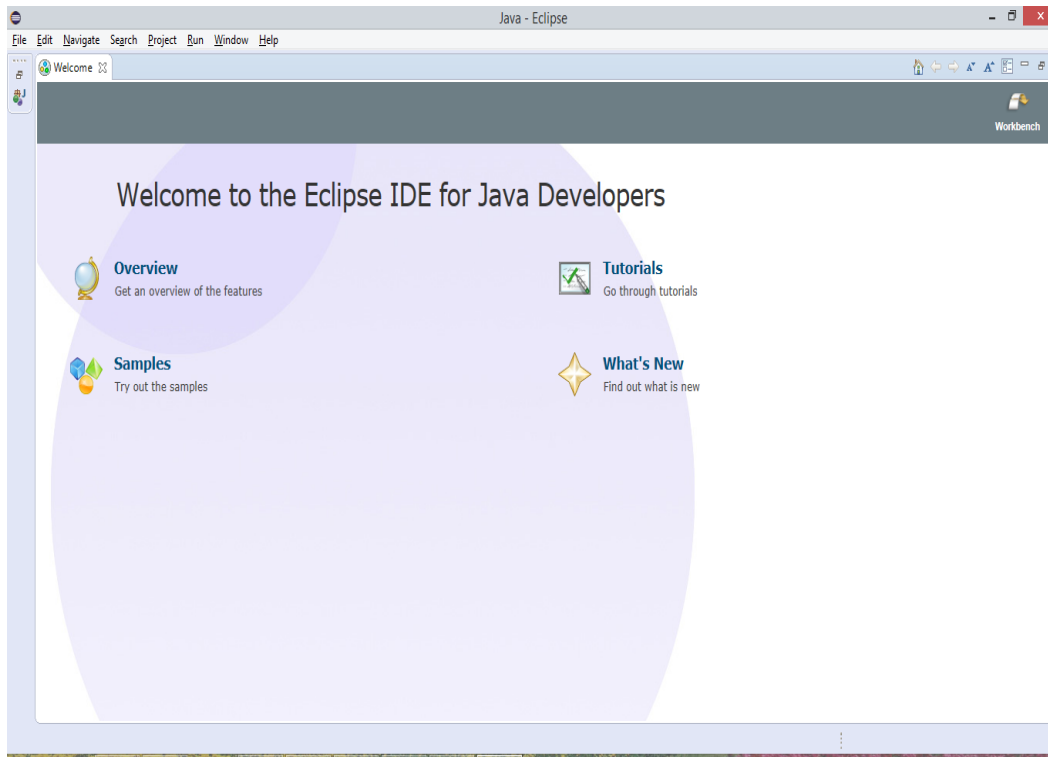
<https://www.sublimetext.com/>



Sublime Text

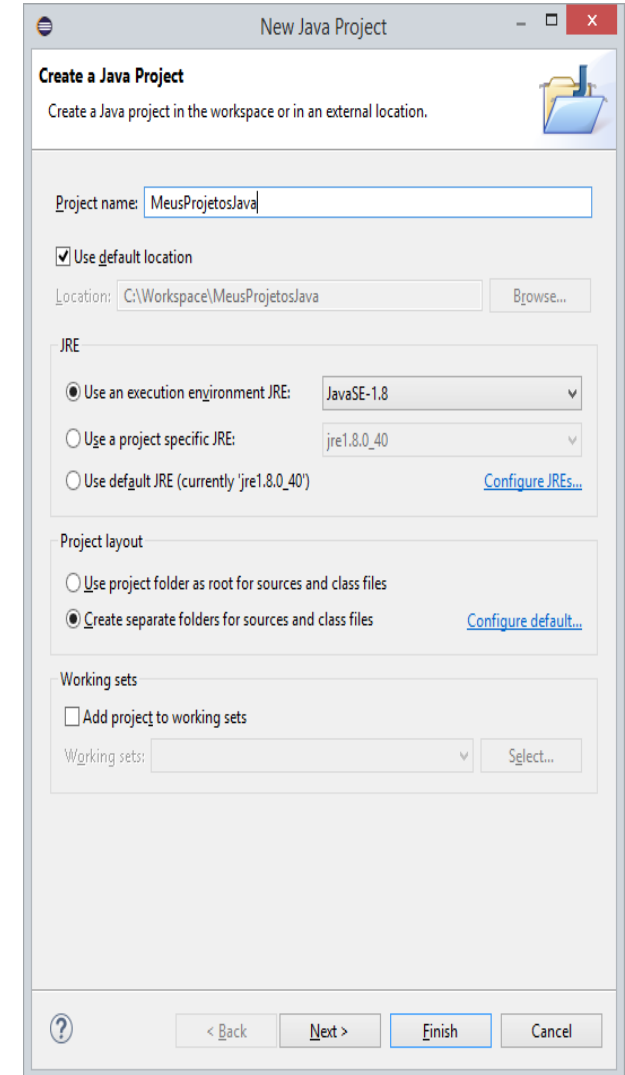
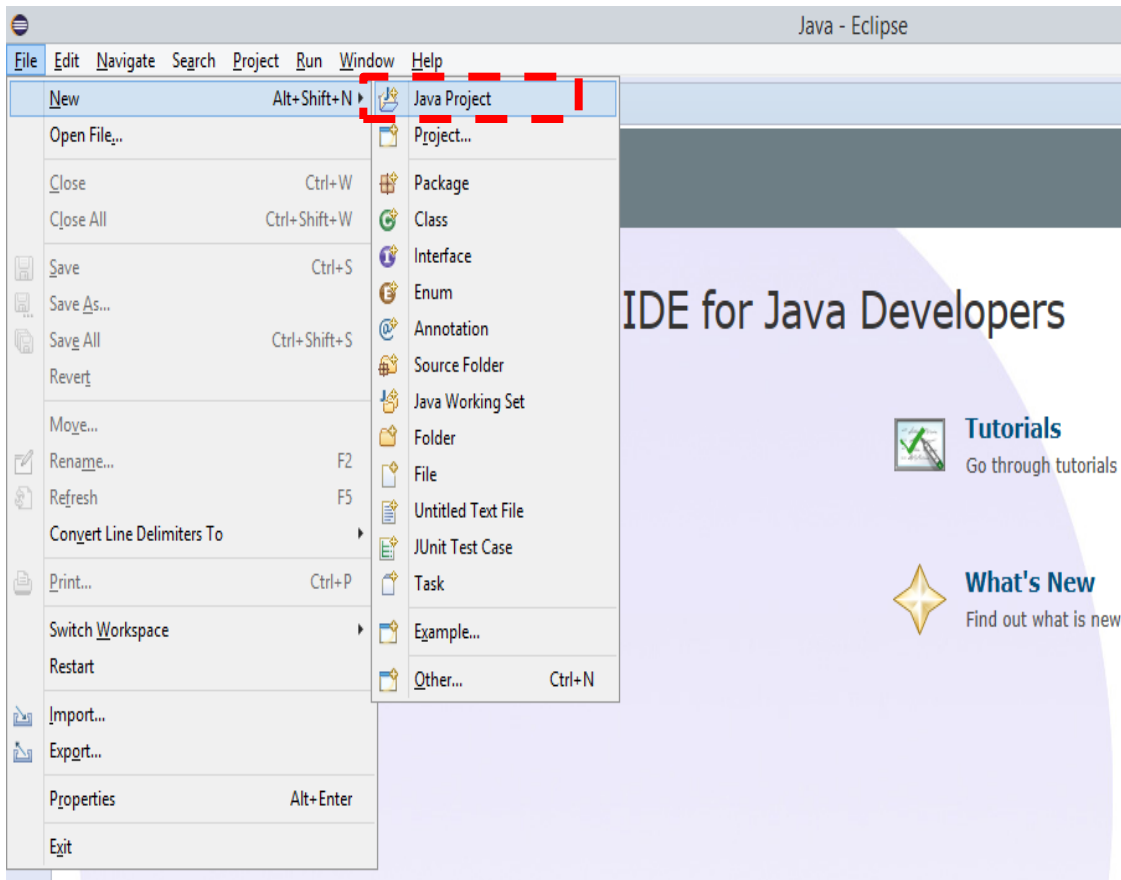
Usando a IDE Eclipse

- Para instalar a IDE basta descompactá-la em alguma pasta. Na primeira execução o Eclipse faz uma breve configuração, tentando localizar e apontar para o diretório do JDK



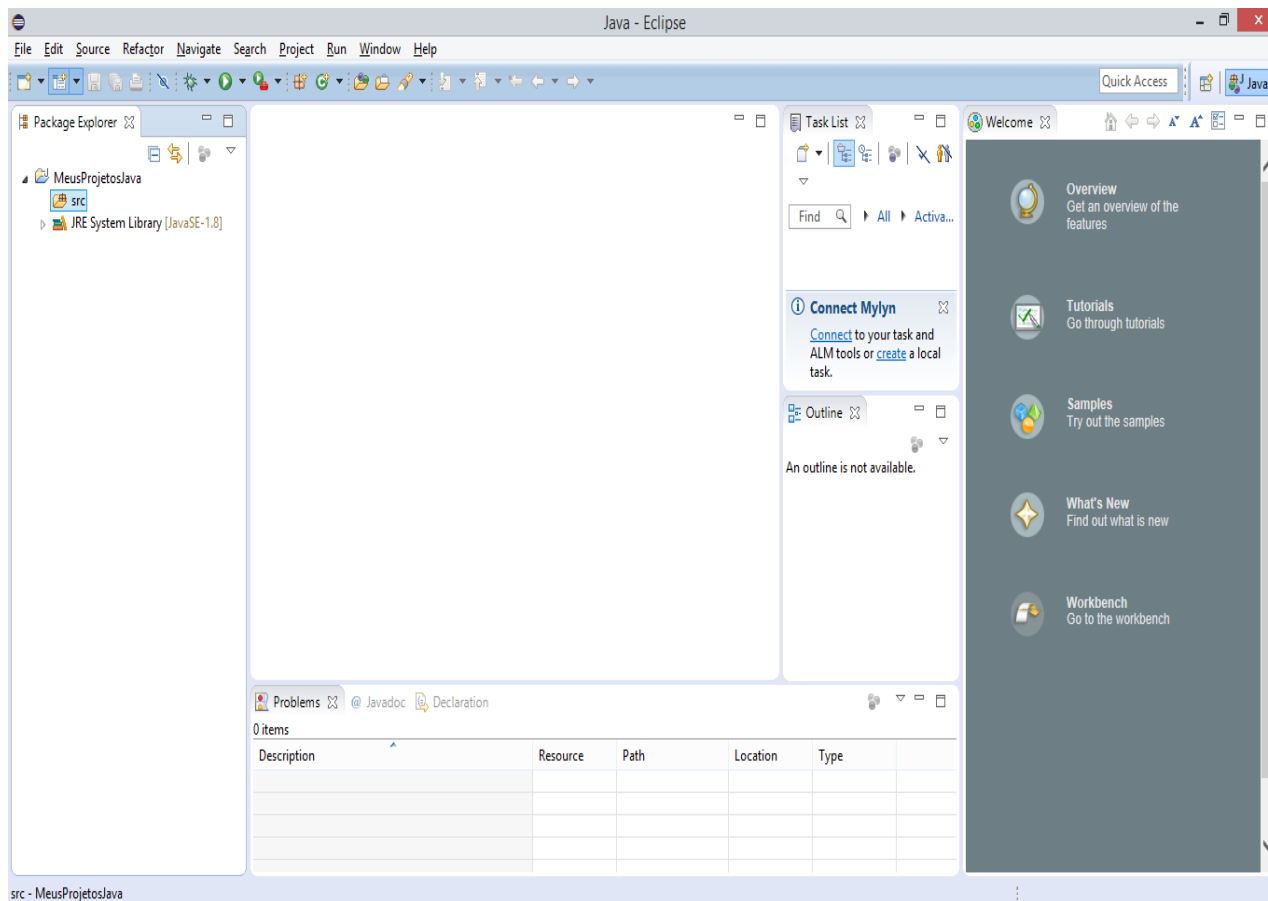
Usando a IDE Eclipse

- Crie um novo projeto Java



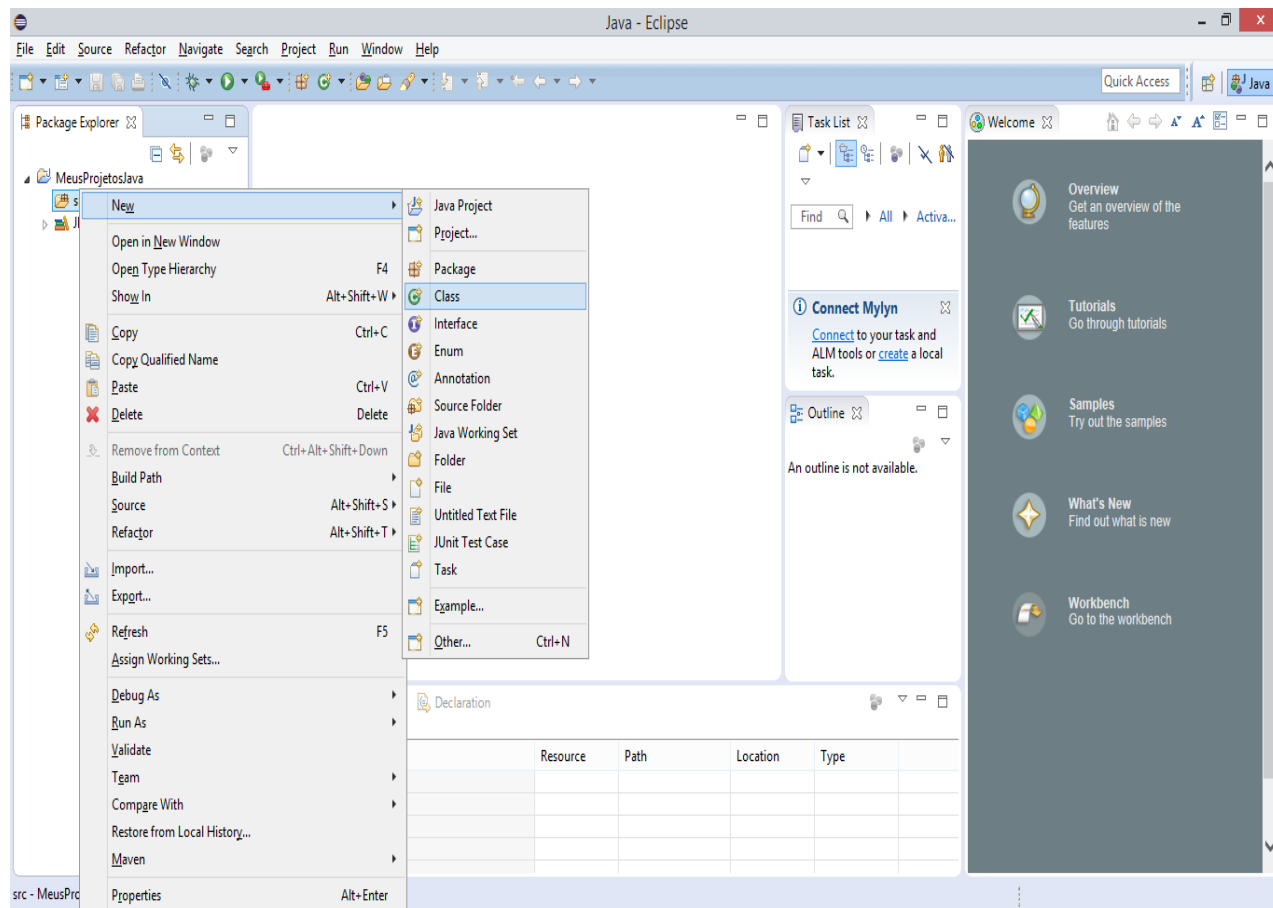
Usando a IDE Eclipse

- Tela inicial do Eclipse com a janela Package Explorer

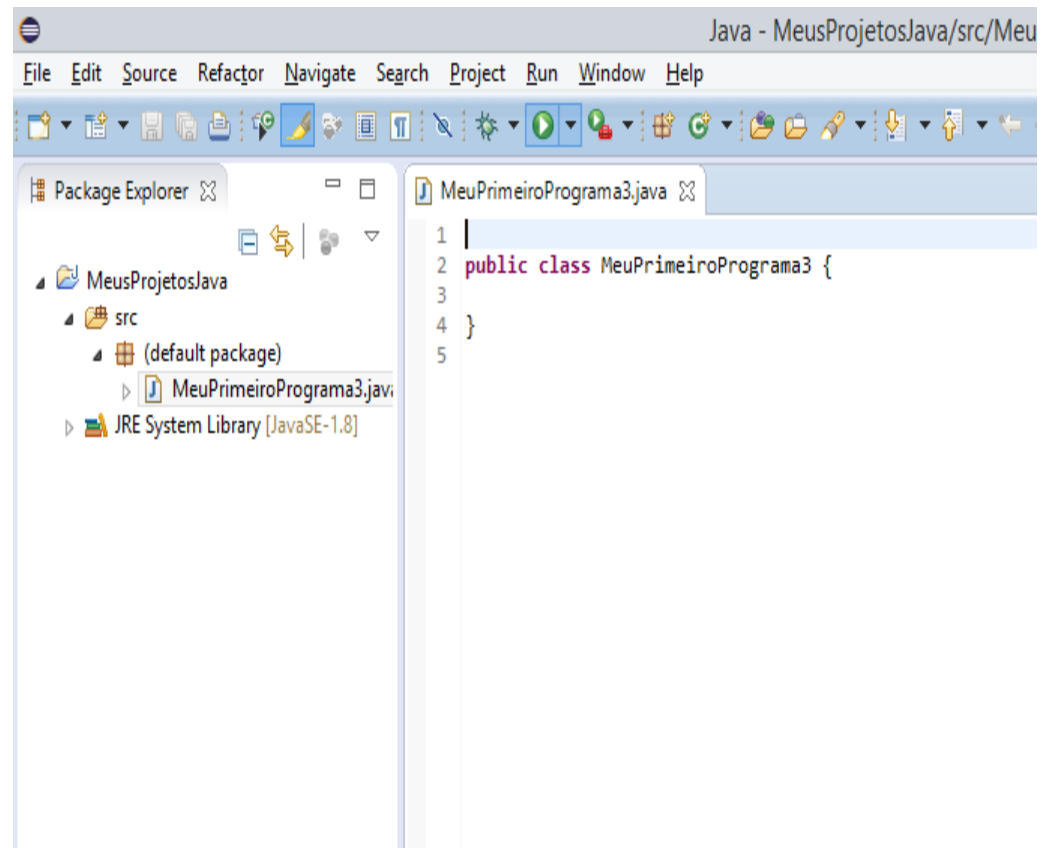
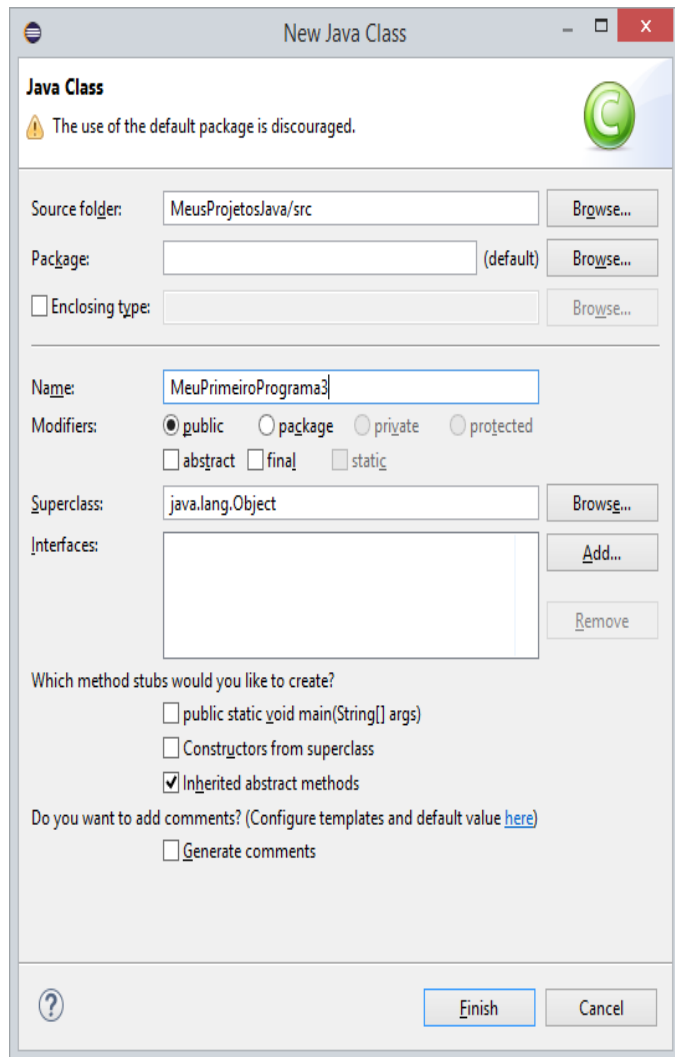


Usando a IDE Eclipse

- Clique no pacote #src com o botão direito e selecione New → Class

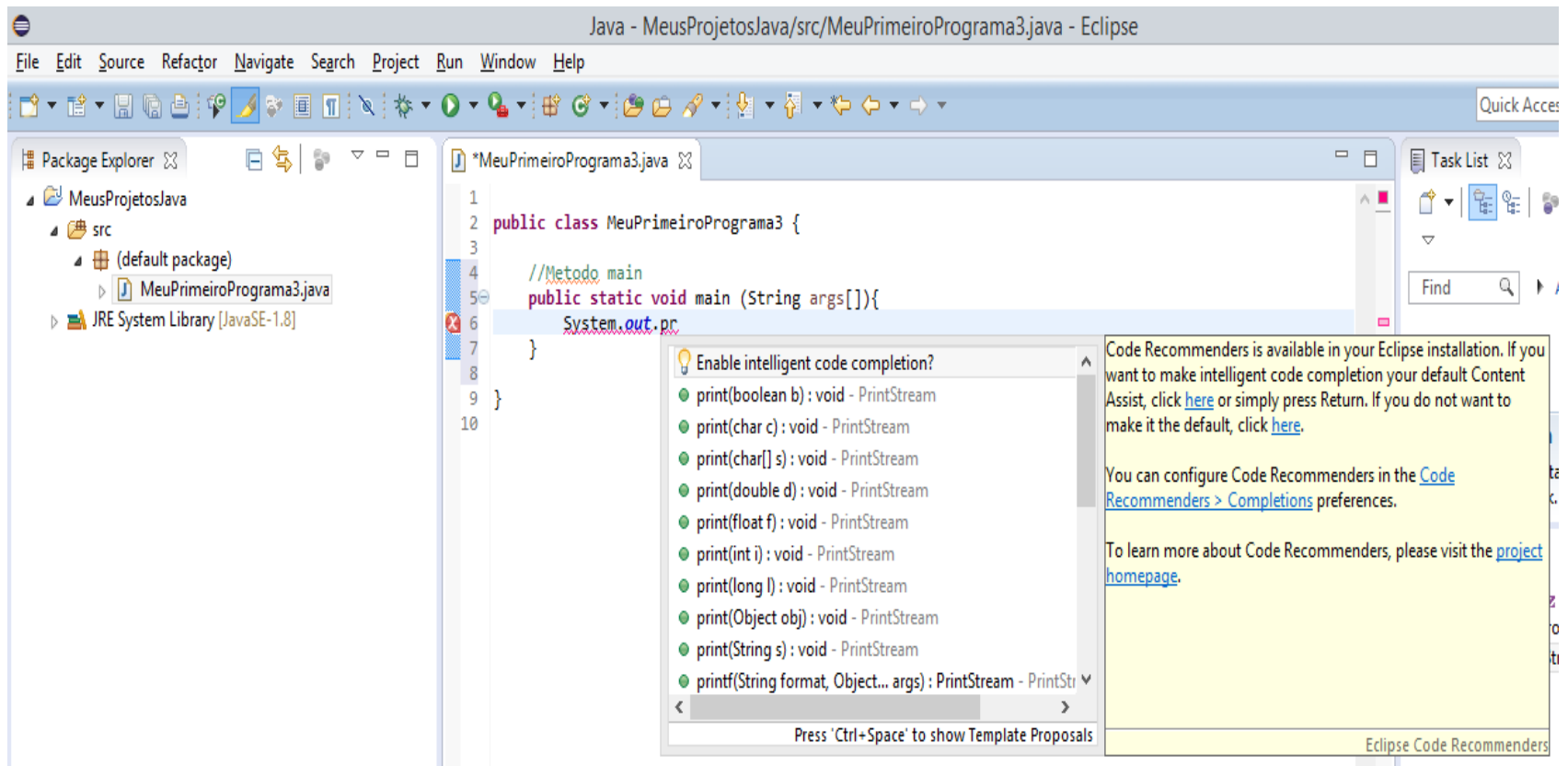


Usando a IDE Eclipse



Usando a IDE Eclipse

- A ferramenta auxilia o programador informando as possibilidades de comandos

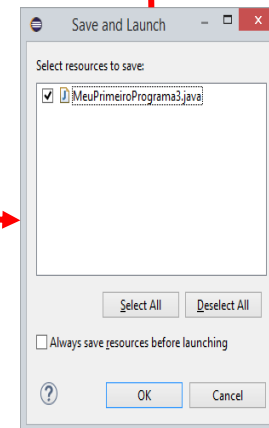
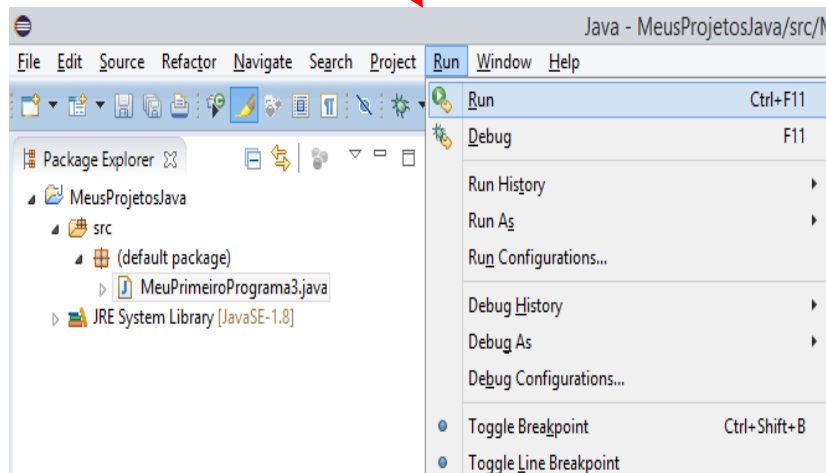


Usando a IDE Eclipse

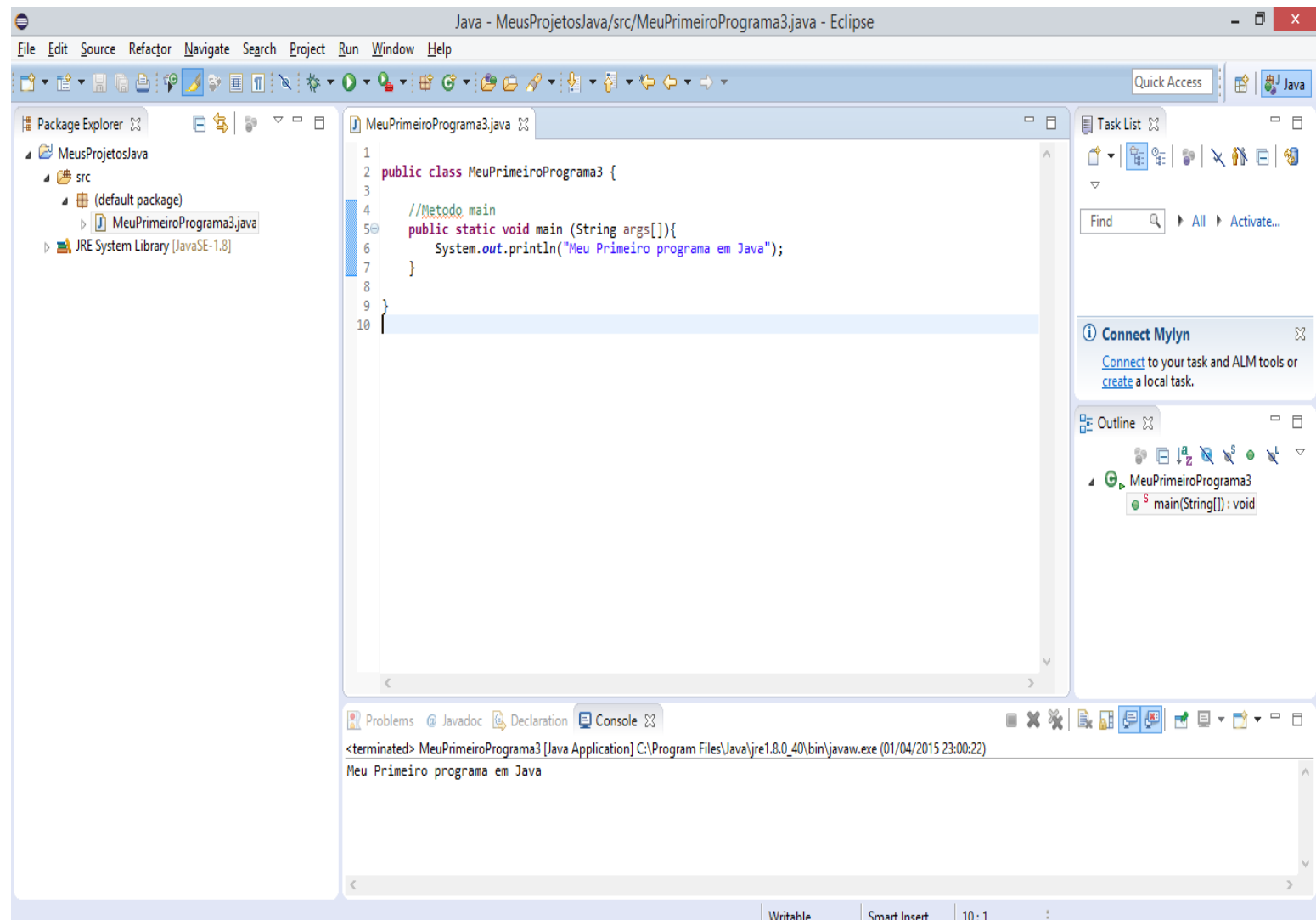
- Compilando e executando o programa

```
*MeuPrimeiroPrograma3.java
1
2 public class MeuPrimeiroPrograma3 {
3
4 //Metodo main
5 public static void main (String args[]){
6     System.out.println("Meu Primeiro programa em Java");
7 }
8
9 }
10 }
```

```
Problems @ Javadoc Declaration Console
<terminated> MeuPrimeiroPrograma3 [Java Application] C:\Program Files\Java\jre1.8.0_40\bin\javaw.exe (01/04/2015 23:00:22)
Meu Primeiro programa em Java
```

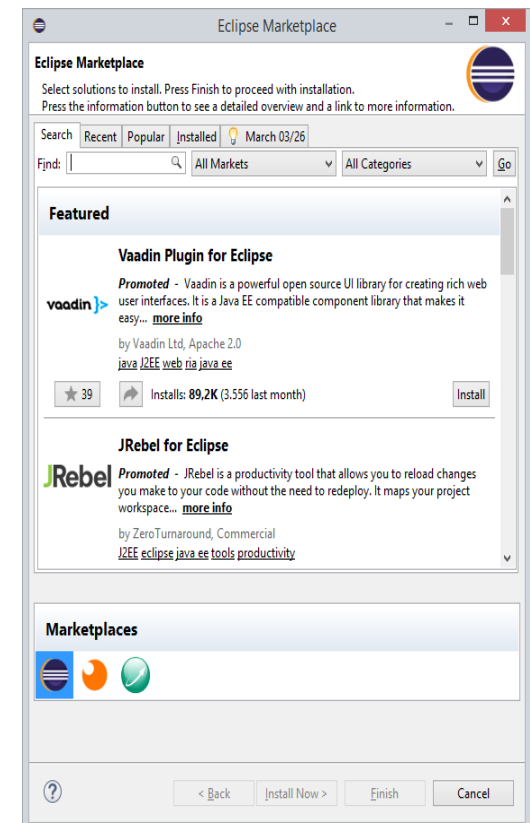
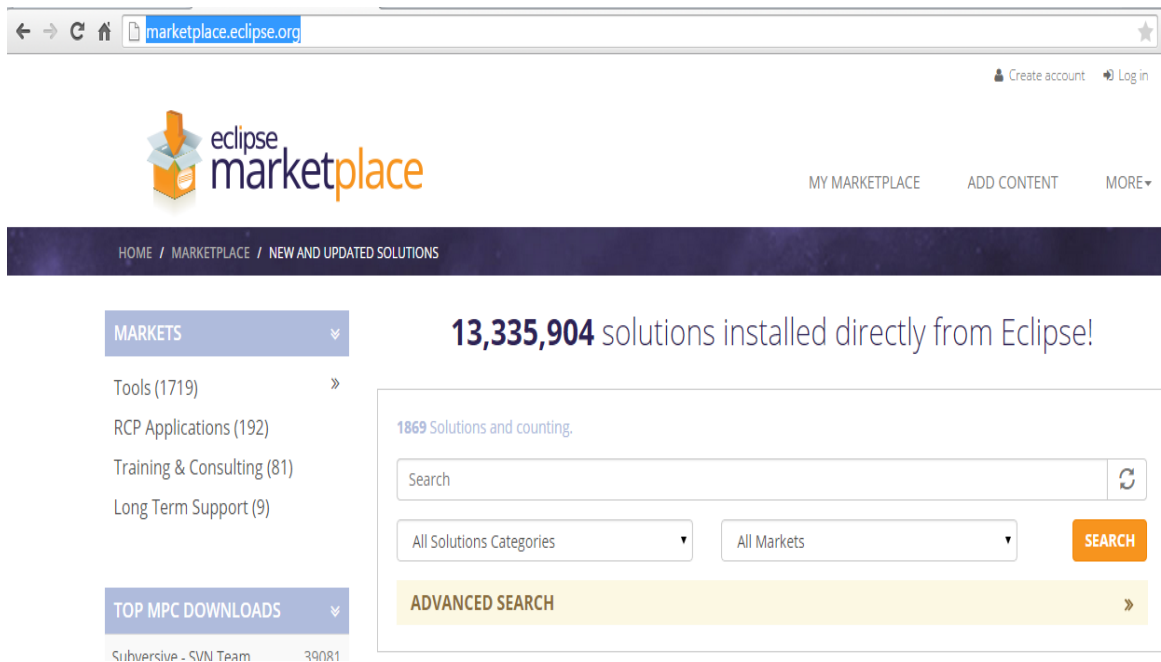


Usando a IDE Eclipse



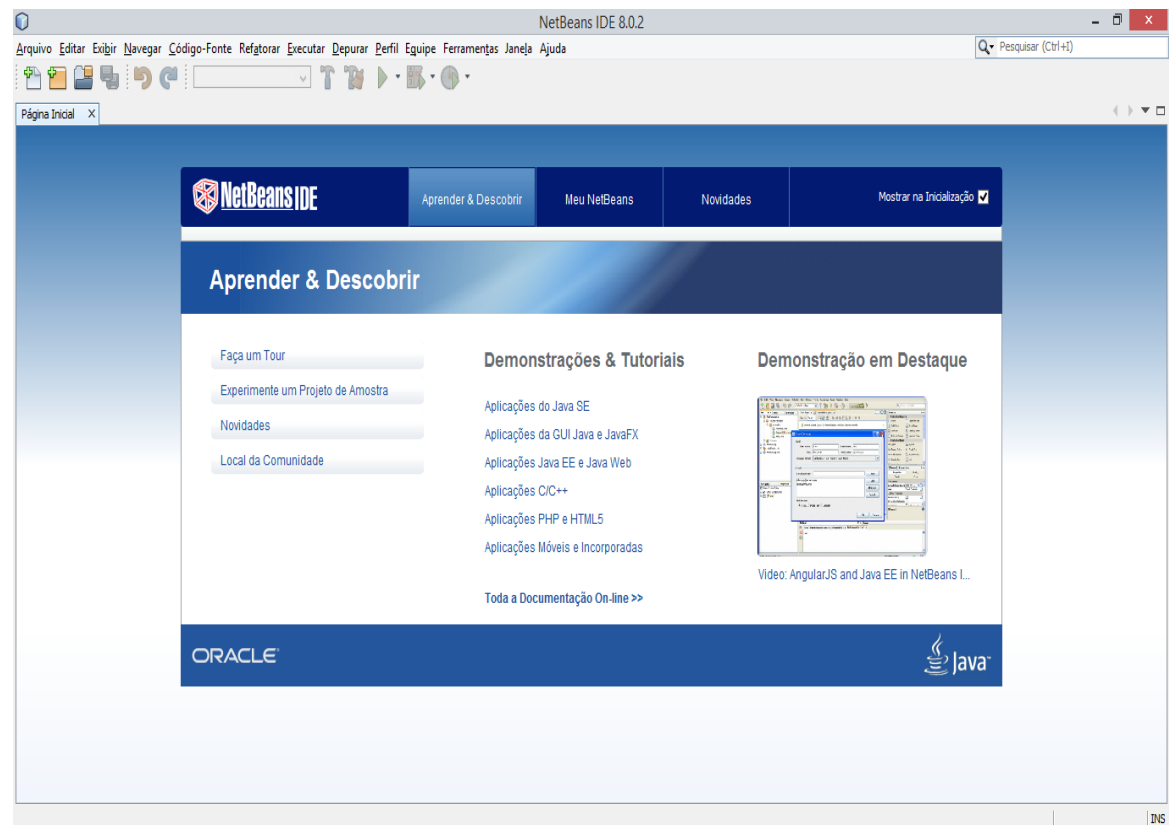
Usando a IDE Eclipse

- O Eclipse possui uma quantidade muito grande de plugins para auxiliar na codificação
- Pelo site <http://marketplace.eclipse.org/> ou na própria IDE



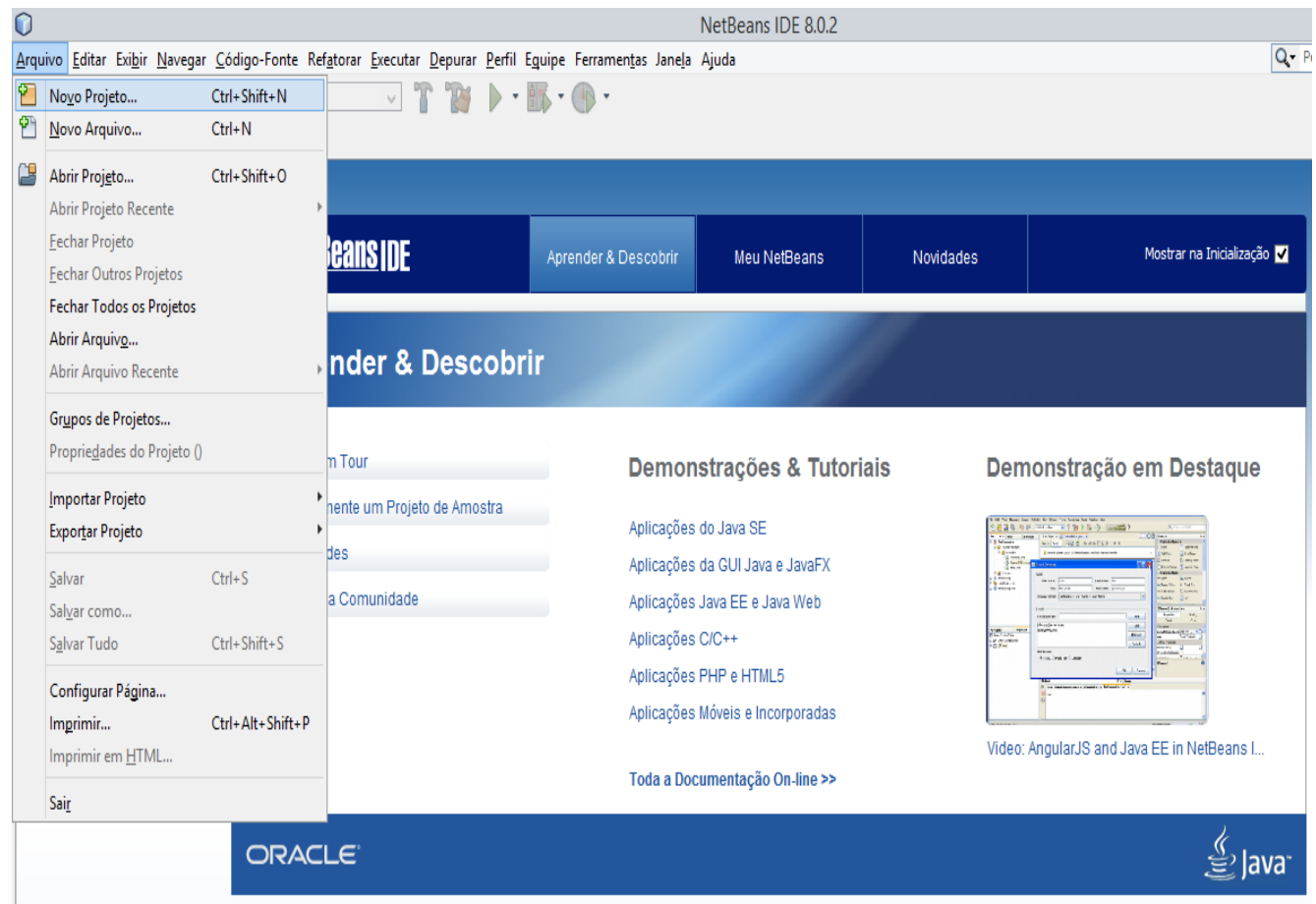
Usando a IDE Netbeans

- Muito similar em termos de funcionalidades ao Eclipse
- Possui um pouco mais de assistentes



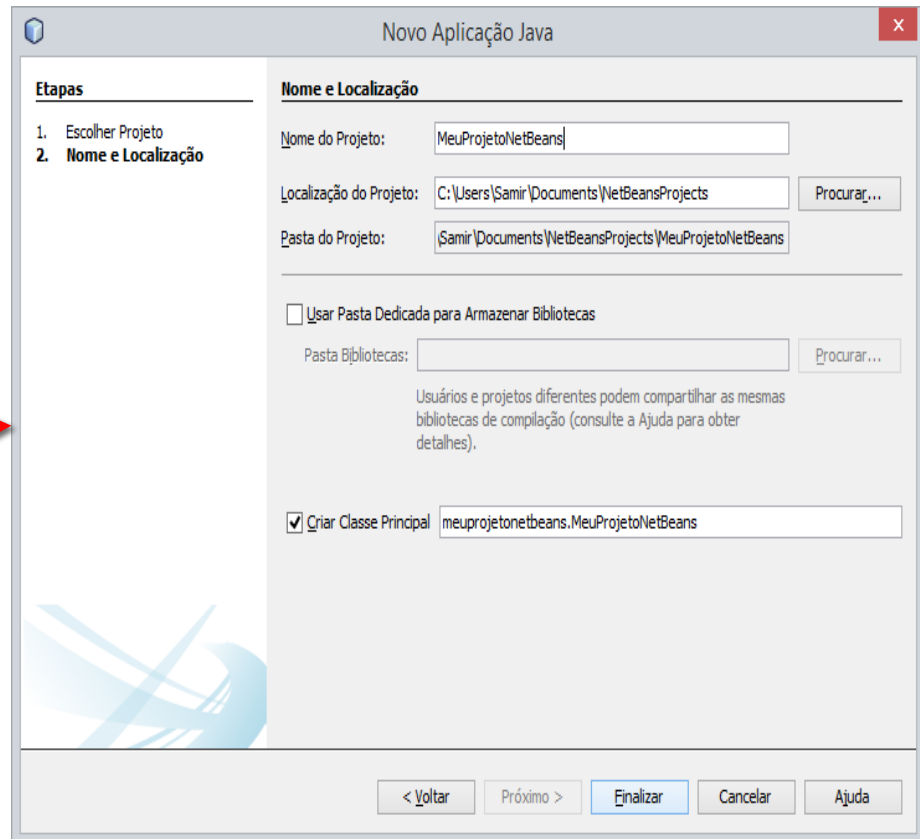
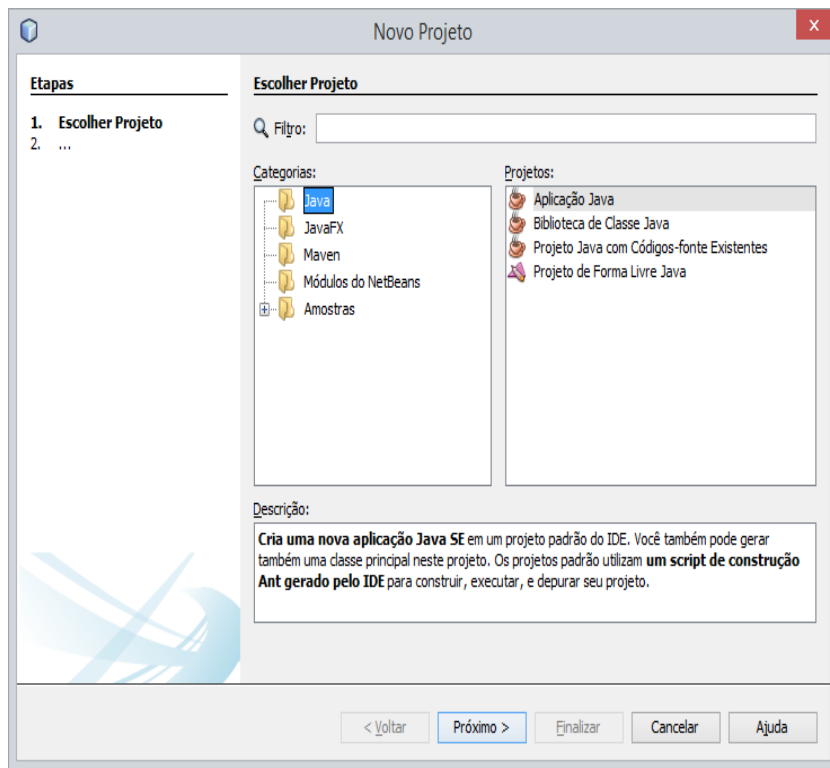
Usando a IDE Netbeans

- Criando um novo projeto

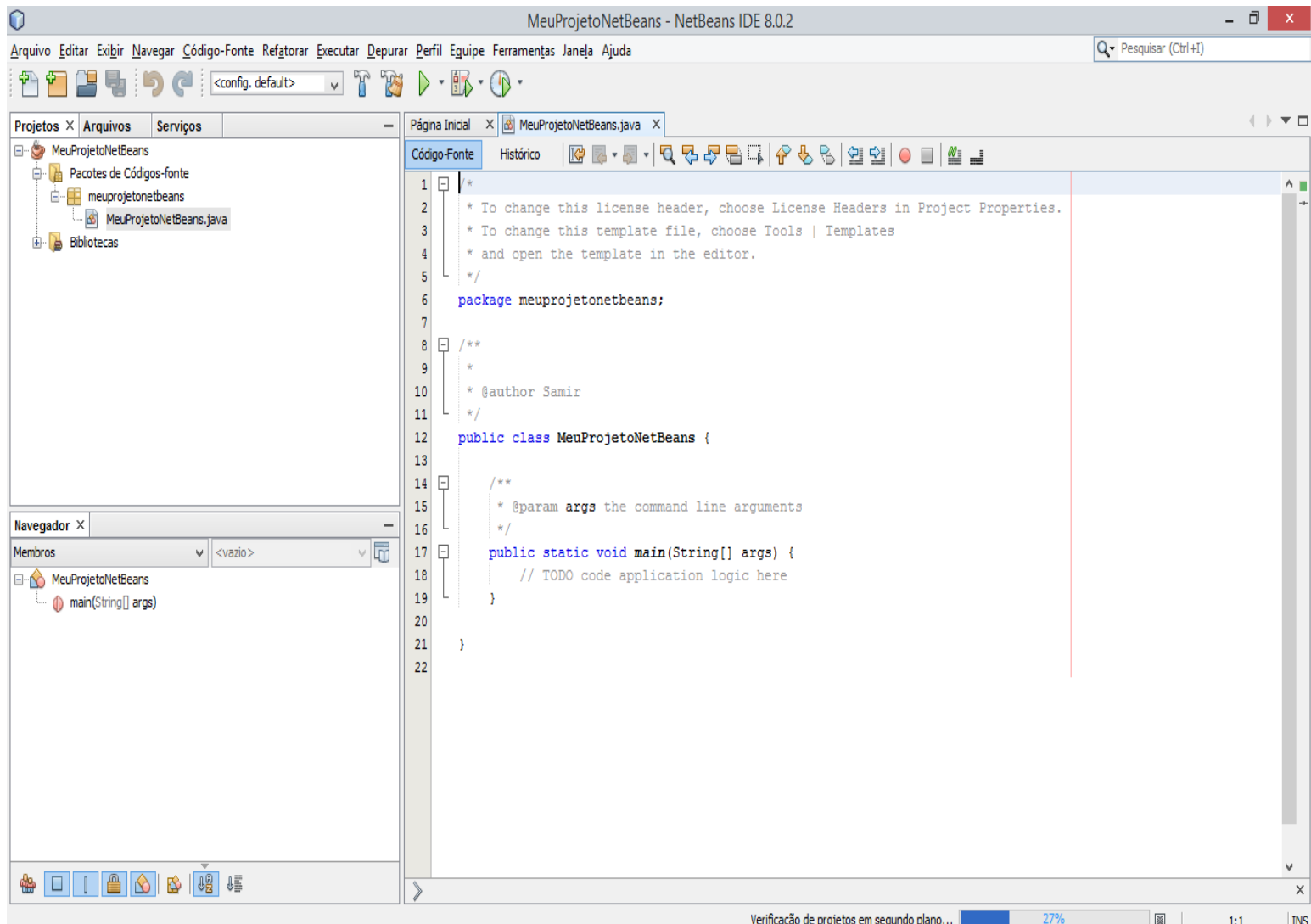


Usando a IDE Netbeans

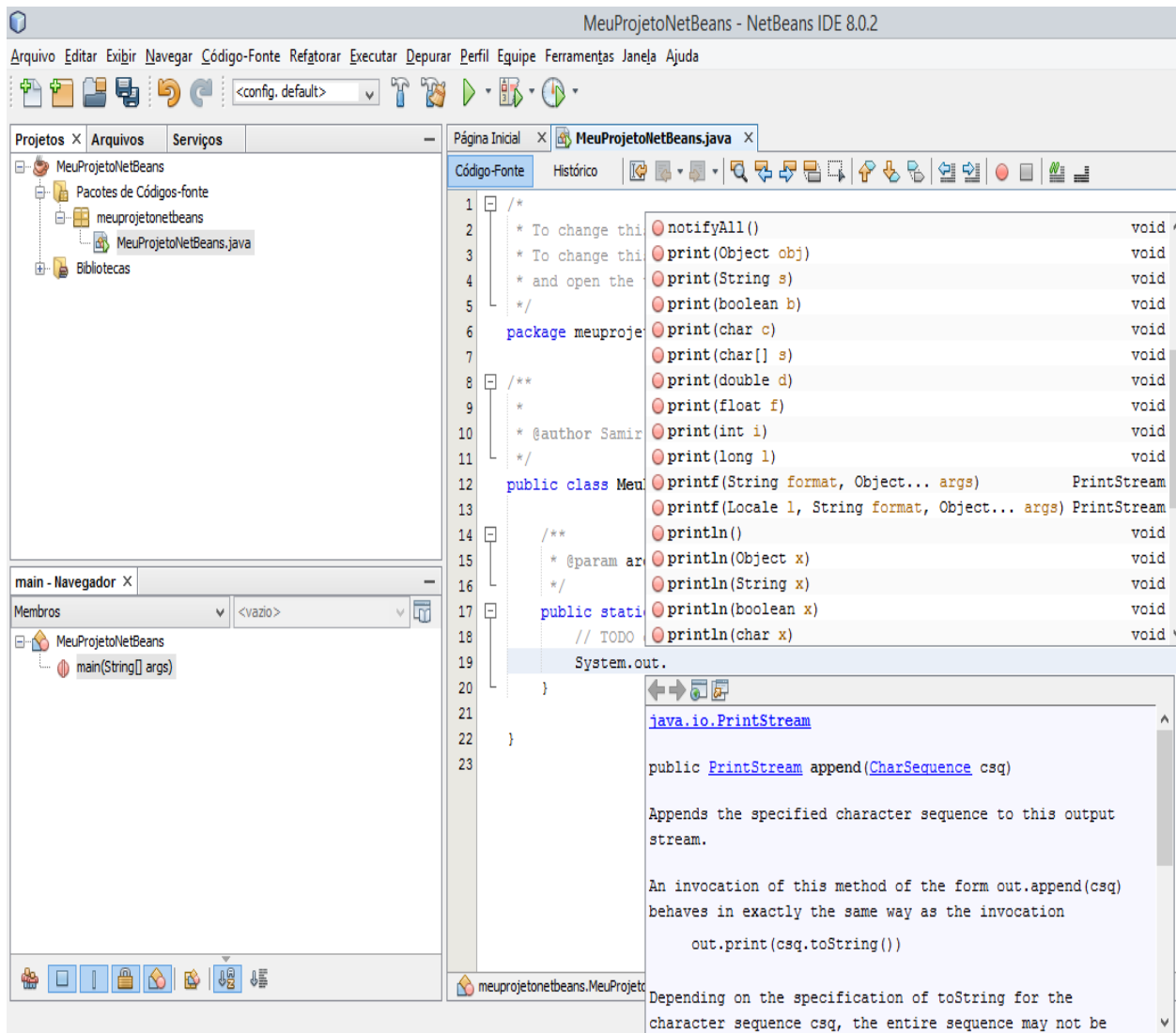
- Criando um novo projeto



Usando a IDE Netbeans



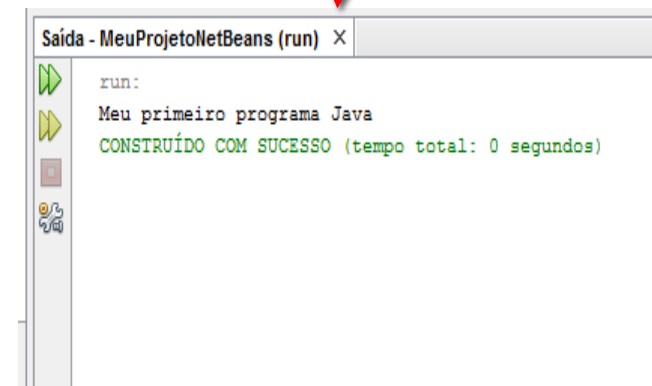
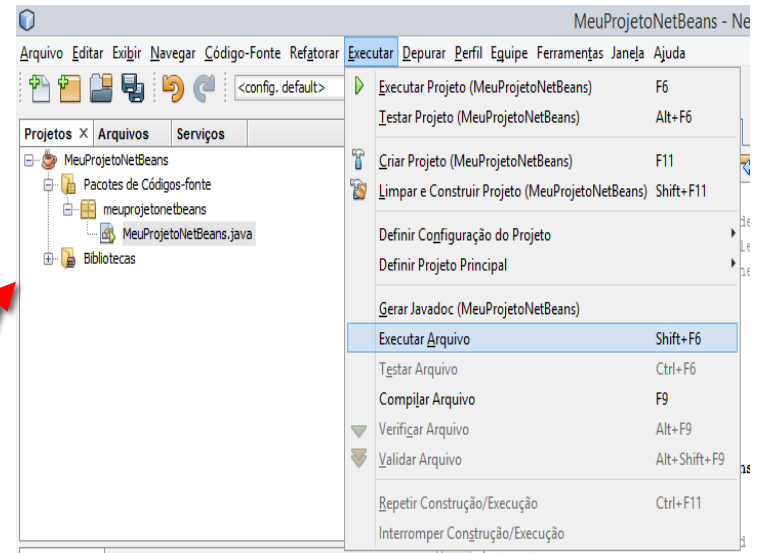
Usando a IDE Netbeans



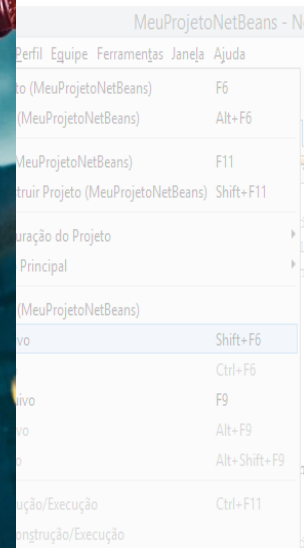
Usando a IDE Netbeans

```
Página Inicial x MeuProjetoNetBeans.java x
Código-Fonte Histórico

1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package meuprojetonetbeans;
7
8  /**
9   *
10   * @author Samir
11   */
12  public class MeuProjetoNetBeans {
13
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("Meu primeiro programa Java");
20      }
21
22  }
23
```



Usando



Página Inicial x MeuProjetoNetBeans.java
Código-Fonte Histórico

```
1  /*  
2   * To change this license header, please go to the file header template  
3   * To change this template file, please go to the file template in the  
4   * and open the template in the editor  
5   */  
6  package meuprojetonetbeans;  
7  
8  /**  
9   *  
10   * @author Samir  
11   */  
12  public class MeuProjetoNetBeans {  
13  
14      /**  
15       * @param args the command line arguments  
16       */  
17      public MeuProjetoNetBeans(String[] args) {  
18          // TODO Auto-generated constructor stub  
19          System.out.println("Meu primeiro programa Java");  
20      }  
21  
22      // TODO Auto-generated method stub  
23  }
```



**Esse chamo de Modo Iron Man
(Tony Stark) de programação**

Vamos digitar!

```
1 //Primeiro programa em Java
2
3 public class MeuPrimeiroPrograma {
4
5     //Metodo main que inicia a execucao do programa em Java
6
7     public static void main ( String args[])
8     {
9
10         System.out.println("Bem vindo ao primeiro programa em Java");
11
12     } //Fim do metodo main
13
14 } //Fim da classe MeuPrimeiroPrograma
```



**KEEP
CALM
AND
HAVE
A BREAK**