



Estácio

Programação I

PROF. LUCAS CAMPOS DE M. NUNES

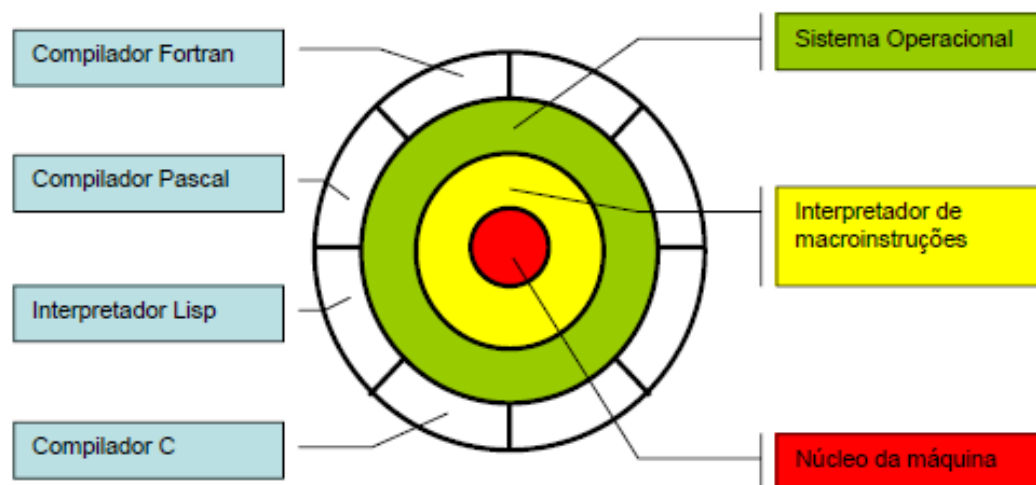
<http://lattes.cnpq.br/2803226406709573>

Brasília-DF, agosto de 2017.



Linguagem de Programação

- Conjunto de regras sintáticas e semânticas utilizados para se estabelecer comunicação entre o programador e o computador.
- São usadas para se escrever programas ou softwares.





Métodos Implementação das LPG's

- É a forma como uma linguagem de programação se comunica (é entendida e executada) com o computador.
 - O computador possui uma linguagem de máquina de nível baixo que oferece operações primitivas.
 - O software de sistema deve criar uma interface com os programas de nível mais alto.
 - O sistema operacional e as implementações de linguagens são dispostos em camadas sobre a interface de linguagem de máquina de um computador.



Introdução ao Java

- Linguagem de programação orientada a objetos desenvolvida por um time de programadores liderados por Patrick Naughton, Mike Sheridan e James Gosling na empresa Sun Microsystems.
- É uma linguagem que é compilada para um “bytecode” que é interpretada por uma máquina virtual – JVM (Java virtual machine).
- Empresa fabricante de computadores, semicondutores e softwares com sede em Santa Clara, Califórnia – EUA. Em abril de 2009, a Oracle Corporation comprou a Sun Microsystems, transformando a indústria de Tecnologia da Informação.
- É a compilação de um programa escrito em Java com o objetivo de ser interpretado pela máquina virtual Java - JVM
- Java Virtual Machine – é um programa carrega e interpreta os aplicativos Java, convertendo os bytecodes em códigos executáveis de máquina.



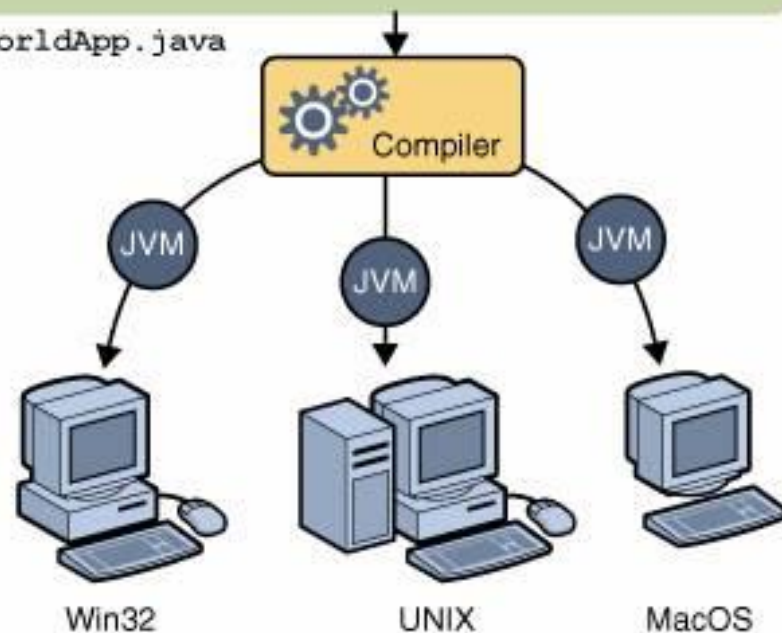
Principais características

- Portabilidade
 - o Java pode ser executado em qualquer plataforma ou equipamento que possua um interpretador Java, e que tenha sido especialmente compilado para o sistema a ser utilizado

Java Program

```
class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

HelloWorldApp.java





Principais características

- Orientada a Objetos – contém a sintaxe similar a linguagem C/C++, mas é baseada no modelo Simula67.
- Segurança – Pode ser executado via rede com restrições de execução, além de proteger o sistema do cliente contra possíveis ataques não-intencionais.



Por que estudar Java

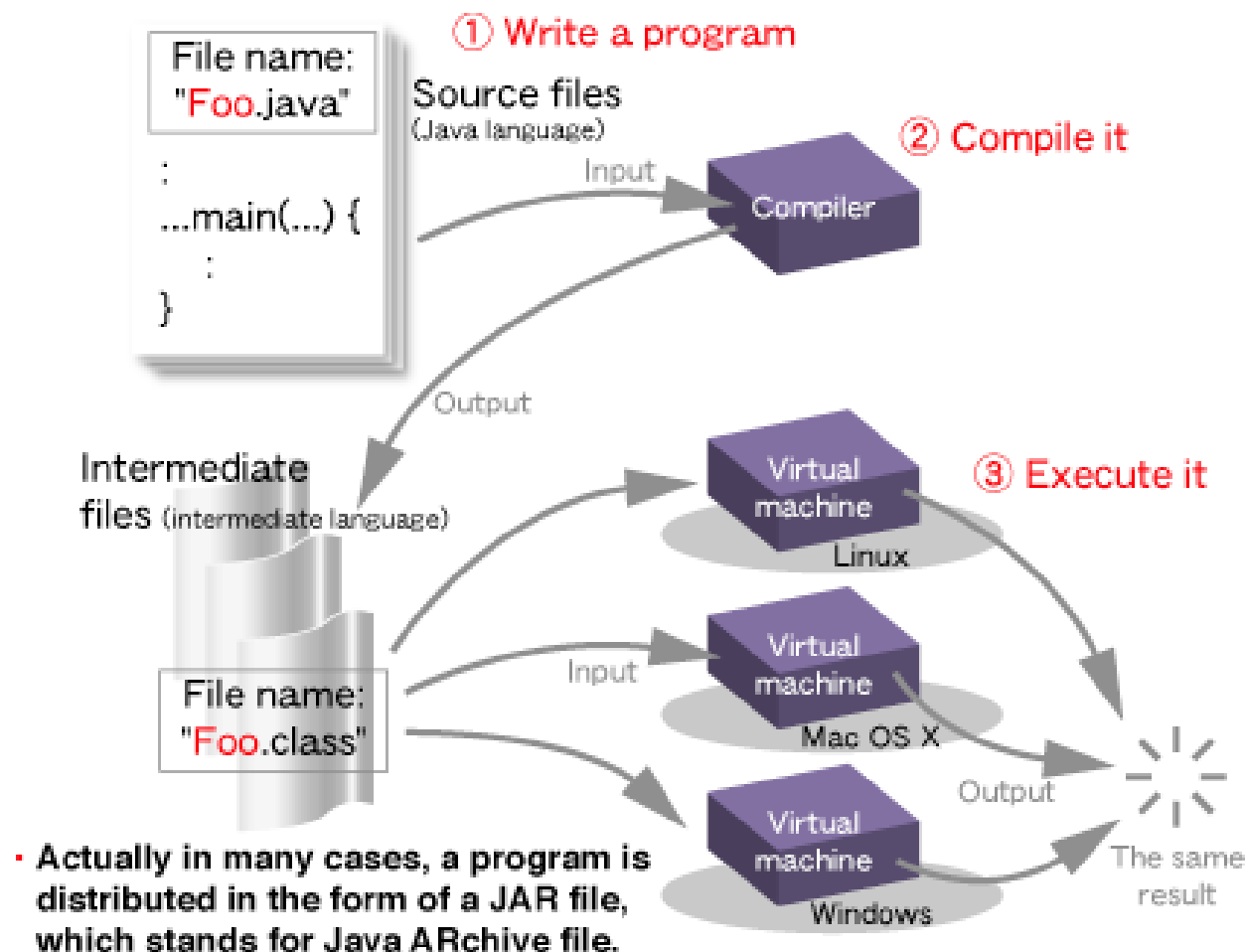
- Orientação a Objetos: Totalmente OO – permitindo herança e reutilização de código de forma dinâmica e estática.
- Dinamismo: permite ser aumentado durante a execução.
- Facilidade: Derivada do C/C++ - Bem Familiar. O ambiente retira do programador a responsabilidade de gerenciar a memória e os ponteiros.

Utilizações e aplicações do Java:

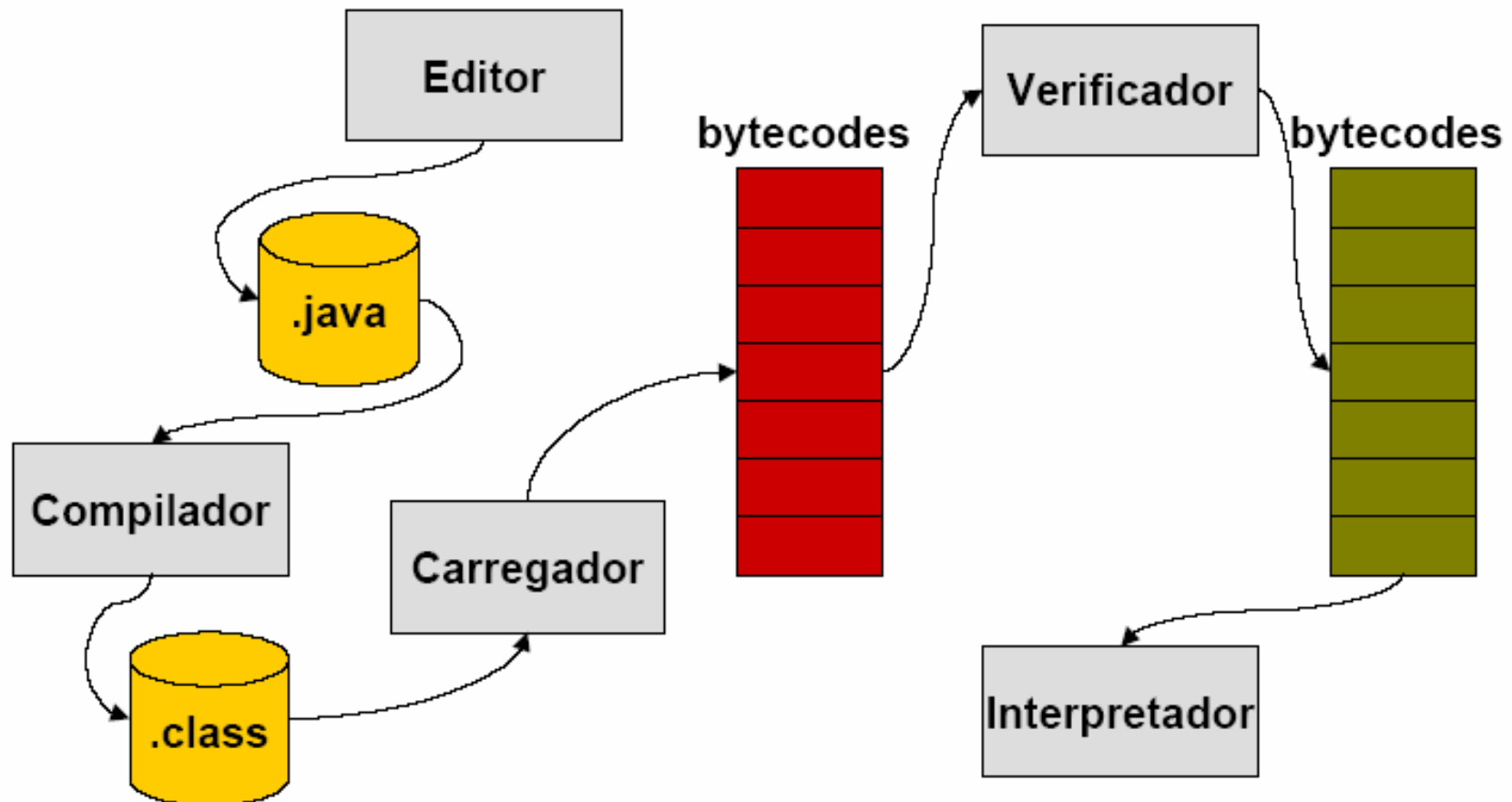
- Desenvolvimento de aplicativos corporativos de grande porte;
- Desenvolvimento de aplicativos Web;
- Fornecimento de aplicativos para dispositivos voltados para o consumo popular (celulares, *paggers*, PDAs, etc);
- Muitos outros propósitos



Resumindo



Funcionamento de um programa Java



Ambiente de programação em Java

- Java SDK (antigamente, JDK): conjunto de aplicativos que permite realizar a compilação e a execução de programas escritos na linguagem Java.
 - o compilador Java, javac,
 - o interpretador de aplicações Java (máquina virtual), java e
 - o interpretador de applets Java, appletviewer.
- Este ambiente pode ser baixado gratuitamente a partir do site da Sun Microsystems <http://java.sun.com> que vai ser direcionado para a página da Oracle:
<http://www.oracle.com/technetwork/java/index.html>

As siglas do Java

- J2SE – Java2 Standard Edition – desenvolvimento de aplicações na arquitetura Cliente – Servidor.
- J2EE – Java2 Enterprise Edition – Fornece um conjunto de APIs para o desenvolvimento corporativo e foca na integração entre sistemas. Disponibiliza alta distribuição de objetos e oferece total suporte a tecnologias para Internet.
- J2ME – Java2 Micro Edition – Fornece as APIs para o desenvolvimento de aplicações para computação móvel, em pequenos dispositivos ou tecnologias embarcadas.



Tipos de programas Java

- Stand – alone – aplicações baseadas na J2SE que tem total acesso aos recursos do sistema, memória, disco, rede, dispositivos, etc.
- Java applets – Pequenas aplicações; não têm acesso aos recursos hardware; necessita navegador com suporte J2SE
- Java servlets – executados em servidores Web, baseados na J2EE, usados para gerar conteúdos dinâmicos-websites.
- Java midlets – pequenos, extremamente seguros e construídos para serem executadas dentro do J2ME.
- JavaBeans – são componentes para serem manipulados visualmente



Construindo um ambiente para programação em Java

- Obter o SDK J2SE apropriado para sua máquina
 - <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Efetuando o Download no site da Oracle

Java SE
Java EE
Java ME
Java SE Support
Java SE Advanced & Suite
Java Embedded
JavaFX
Java DB
Web Tier
Java Card
Java TV
New to Java
Community
Java Magazine

- Overview
- Downloads**
- Documentation
- Community
- Technologies
- Training

Java SE Downloads

[Latest Release](#)
[Next Release \(Early Access\)](#)
[Embedded Use](#)
[Previous Releases](#)


Java Platform (JDK) 7



JavaFX 2.0 Beta



JDK 7 + NetBeans Bundle



JDK 7 + Java EE Bundle

Here are the Java SE downloads in detail:

Java Platform, Standard Edition

Java SE 7

This release includes new features such as small language changes for improved developer productivity, a new Filesystem API, support for asynchronous I/O, a new fork/join framework for multicore performance, improved support for dynamic and script languages, updates to security, internationalization and web standards and much more.

[Learn more](#) ▶

JDK

[Download](#)

JDK 7 Docs

- [Installation Instructions](#)
- [ReadMe](#)
- [ReleaseNotes](#)

JRE

[Download](#)

JRE 7 Docs

- [Installation Instructions](#)
- [ReadMe](#)
- [ReleaseNotes](#)

Java SDKs and Tools

- ↓ [Java SE](#)
- ↓ [Java EE and Glassfish](#)
- ↓ [Java ME](#)
- ↓ [JavaFX](#)
- ↓ [Java Card](#)
- ↓ [NetBeans IDE](#)

Java Resources

- ↓ [New to Java?](#)
- ↓ [APIs](#)
- ↓ [Code Samples & Apps](#)
- ↓ [Developer Training](#)
- ↓ [Documentation](#)
- ↓ [Java BluePrints](#)
- ↓ [Java.com](#)
- ↓ [Java.net](#)
- ↓ [Student Developers](#)
- ↓ [Tutorials](#)



Efetuando o Download no site da Oracle

Java SE
Java EE
Java ME
Java SE Support
Java SE Advanced & Suite
Java Embedded
JavaFX
Java DB
Web Tier
Java Card
Java TV
New to Java
Community
Java Magazine

- Overview
- Downloads**
- Documentation
- Community
- Technologies
- Training

Java SE Development Kit 7 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java™ platform.

Java SE Development Kit 7

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

☐ Accept License Agreement
 ☒ Decline License Agreement

Product / File Description	File Size	Download
Linux x86 - RPM Installer	77.28 MB	jdk-7-linux-i586.rpm
Linux x86 - Compressed Binary	92.17 MB	jdk-7-linux-i586.tar.gz
Linux x64 - RPM Installer	77.91 MB	jdk-7-linux-x64.rpm
Linux x64 - Compressed Binary	90.57 MB	jdk-7-linux-x64.tar.gz
Solaris x86 - Compressed Packages	154.74 MB	jdk-7-solaris-i586.tar.gz
Solaris x86 - Compressed Binary	94.75 MB	jdk-7-solaris-i586.tar.gz
Solaris SPARC - Compressed Packages	157.81 MB	jdk-7-solaris-sparc.tar.gz
Solaris SPARC - Compressed Binary	99.48 MB	jdk-7-solaris-sparc.tar.gz
Solaris SPARC 64-bit - Compressed Packages	16.28 MB	jdk-7-solaris-sparcv9.tar.gz
Solaris SPARC 64-bit - Compressed Binary	12.38 MB	jdk-7-solaris-sparcv9.tar.gz
Solaris x64 - Compressed Packages	14.66 MB	jdk-7-solaris-x64.tar.gz
Solaris x64 - Compressed Binary	9.39 MB	jdk-7-solaris-x64.tar.gz
Windows x86	79.48 MB	jdk-7-windows-i586.exe
Windows x64	80.25 MB	jdk-7-windows-x64.exe

Java SDKs and Tools

- [Java SE](#)
- [Java EE and Glassfish](#)
- [Java ME](#)
- [JavaFX](#)
- [Java Card](#)
- [NetBeans IDE](#)

Java Resources

- [New to Java?](#)
- [APIs](#)
- [Code Samples & Apps](#)
- [Developer Training](#)
- [Documentation](#)
- [Java BluePrints](#)
- [Java.com](#)
- [Java.net](#)
- [Student Developers](#)
- [Tutorials](#)





Conferir a instalação do Java

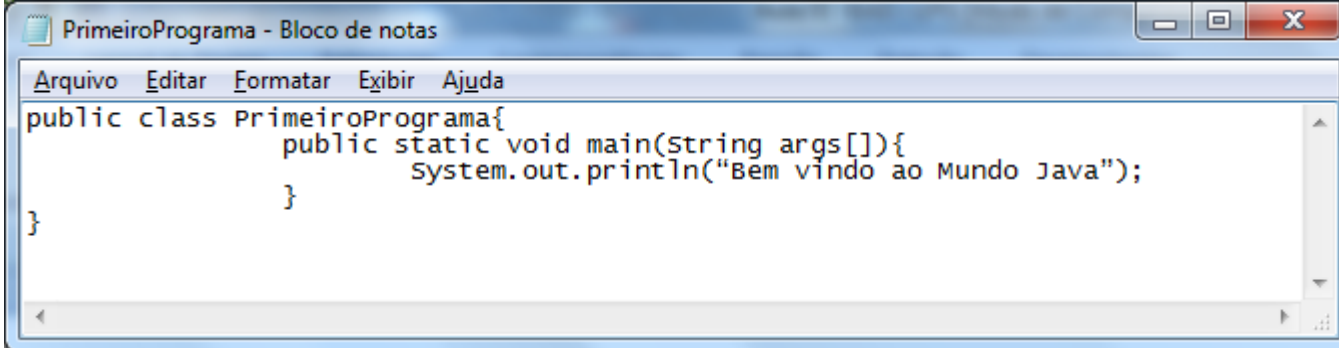
```
C:\Users\Eduardo>javac
Usage: javac <options> <source files>
where possible options include:
  -g               Generate all debugging info
  -g:none          Generate no debugging info
  -g:{lines,vars,source}  Generate only some debugging info
  -nowarn          Generate no warnings
  -verbose         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
  -d <directory>     Specify where to place generated class files
  -s <directory>     Specify where to place generated source 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
  -version           Version information
  -help             Print a synopsis of standard options
  -Akey[=value]     Options to pass to annotation processors
  -X               Print a synopsis of nonstandard options
  -J<flag>          Pass <flag> directly to the runtime system

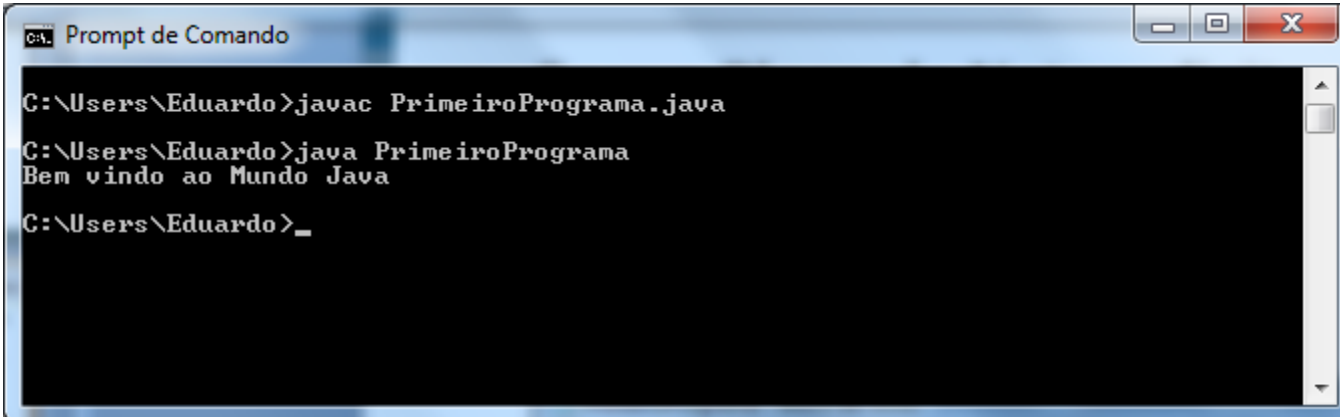
C:\Users\Eduardo>_
```

Escrevendo e executando o primeiro programa

- Com o Bloco de Notas, digitar o texto.
- Gravar a classe com o mesmo nome do arquivo .java



```
Arquivo  Editar  Formatar  Exibir  Ajuda
public class PrimeiroPrograma{
    public static void main(String args[]){
        System.out.println("Bem vindo ao Mundo Java");
    }
}
```



```
C:\Users\Eduardo>javac PrimeiroPrograma.java
C:\Users\Eduardo>java PrimeiroPrograma
Bem vindo ao Mundo Java
C:\Users\Eduardo>_
```



Parabéns

- Acabamos de executar nosso primeiro programa.
- Observação: O compilador java e a sua máquina virtual, foram instalados dentro da pasta 'bin' da pasta da versão do JDK escolhida. Para que você consiga chamar estes programas de qualquer parte do seu computador, você terá que incluir o caminho da pasta 'bin' na variável de ambiente do seu sistema chamada 'Path'.

Configurando o Path

