dLab 1 – Introdução a Linguagem Java

Neste laboratório faremos Instalação do **Java Development Kit**, em ambiente Windows ou Linux. E escreveremos nosso primeiro programa em Java, o mais famoso programa do mundo.

Exercícios

Exercício 1: Instalar JDK 13 (30 minutos)

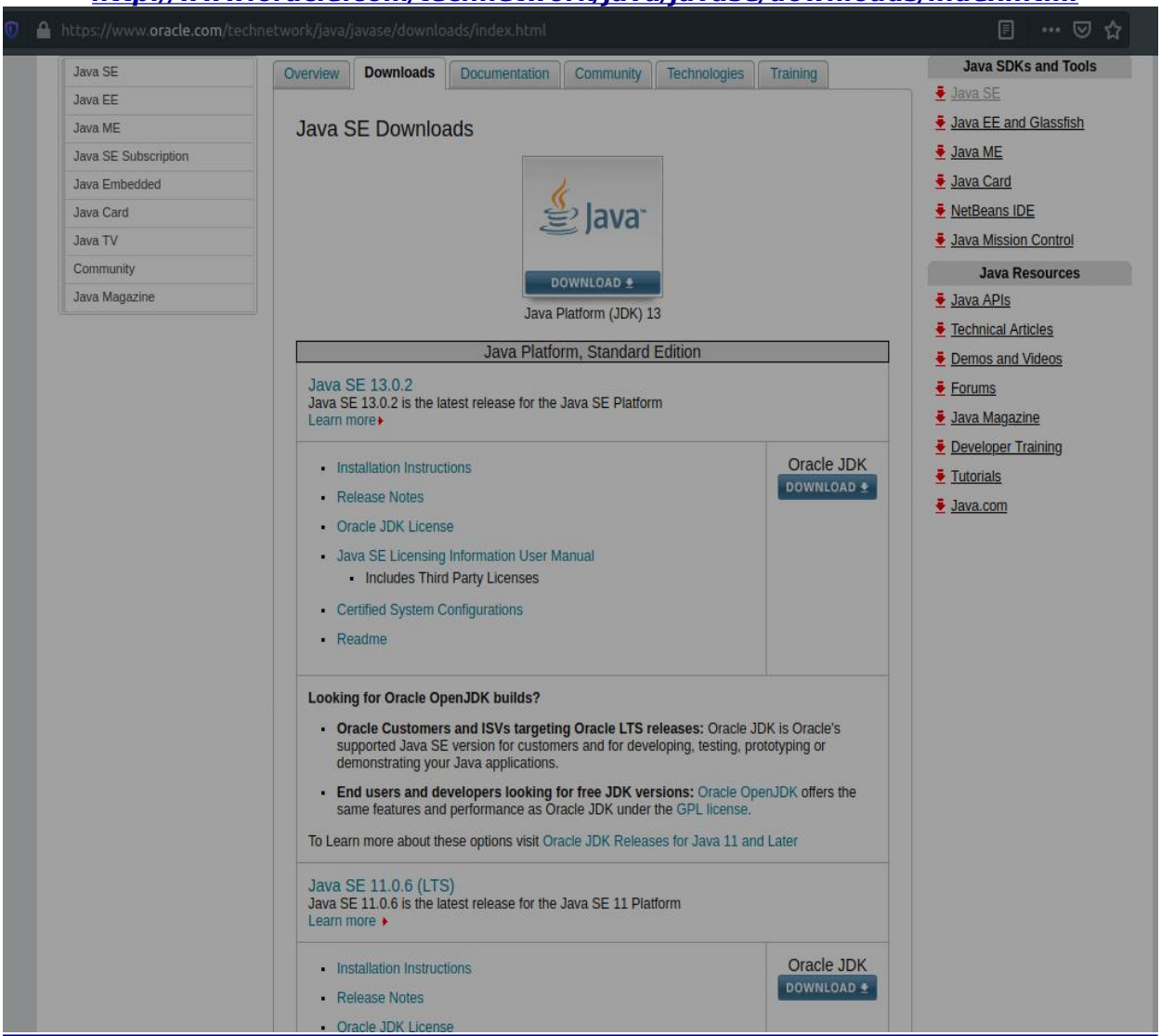
Exercício 2: Escrever, compilar e rodar um programa **HelloWorld** em Java usando ferramentas da linha de comando (30 minutos)

Exercício 1 - Instalar JDK 13

1.1 – Fazendo Download do JDK 13

Para você baixar o JDK(Java Development Kit), acesse o link no site da ORACLE:

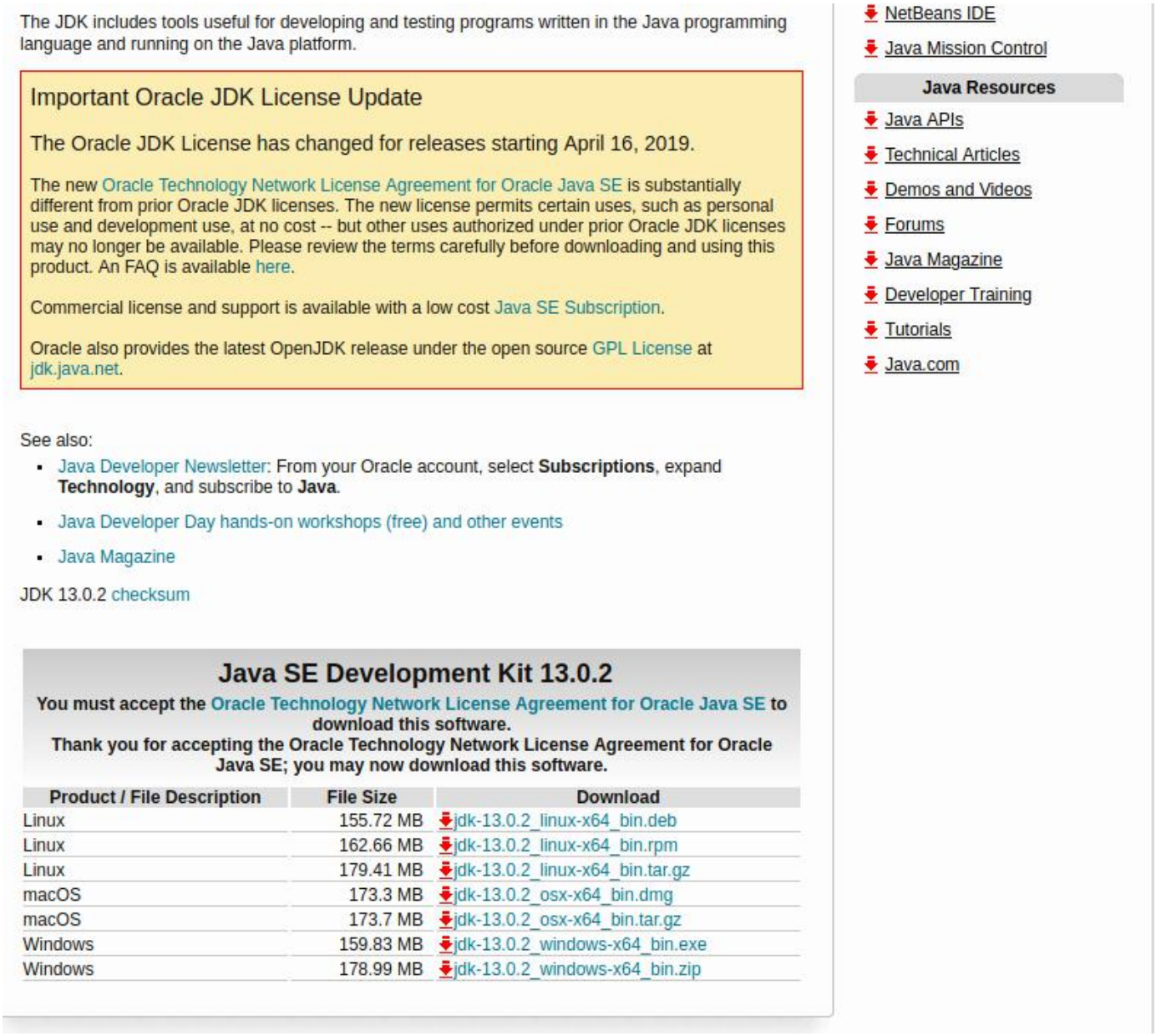
http://www.oracle.com/technetwork/java/javase/downloads/index.html



Nesta página, você tem a opção de baixar o **JDK, JRE, Documentação**, **Código Fonte** e

outras.

1. Clique em **Download (Java Plataform JDK 13)** como mostra a figura acima.
2. Você será redirecionado para a página abaixo.



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

Important Oracle JDK License Update

The Oracle JDK License has changed for releases starting April 16, 2019.

The new Oracle Technology Network License Agreement for Oracle Java SE is substantially different from prior Oracle JDK licenses. The new license permits certain uses, such as personal use and development use, at no cost -- but other uses authorized under prior Oracle JDK licenses may no longer be available. Please review the terms carefully before downloading and using this product. An FAQ is available here.

Commercial license and support is available with a low cost Java SE Subscription.

Oracle also provides the latest OpenJDK release under the open source GPL License at jdk.java.net.

See also:

- Java Developer Newsletter: From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK 13.0.2 checksum

Java SE Development Kit 13.0.2

You must accept the Oracle Technology Network License Agreement for Oracle Java SE to download this software.
Thank you for accepting the Oracle Technology Network License Agreement for Oracle Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux	155.72 MB	jdk-13.0.2_linux-x64_bin.deb
Linux	162.66 MB	jdk-13.0.2_linux-x64_bin.rpm
Linux	179.41 MB	jdk-13.0.2_linux-x64_bin.tar.gz
macOS	173.3 MB	jdk-13.0.2_osx-x64_bin.dmg
macOS	173.7 MB	jdk-13.0.2_osx-x64_bin.tar.gz
Windows	159.83 MB	jdk-13.0.2_windows-x64_bin.exe
Windows	178.99 MB	jdk-13.0.2_windows-x64_bin.zip

NetBeans IDE
Java Mission Control

Java Resources

Java APIs
Technical Articles
Demos and Videos
Forums
Java Magazine
Developer Training
Tutorials
Java.com

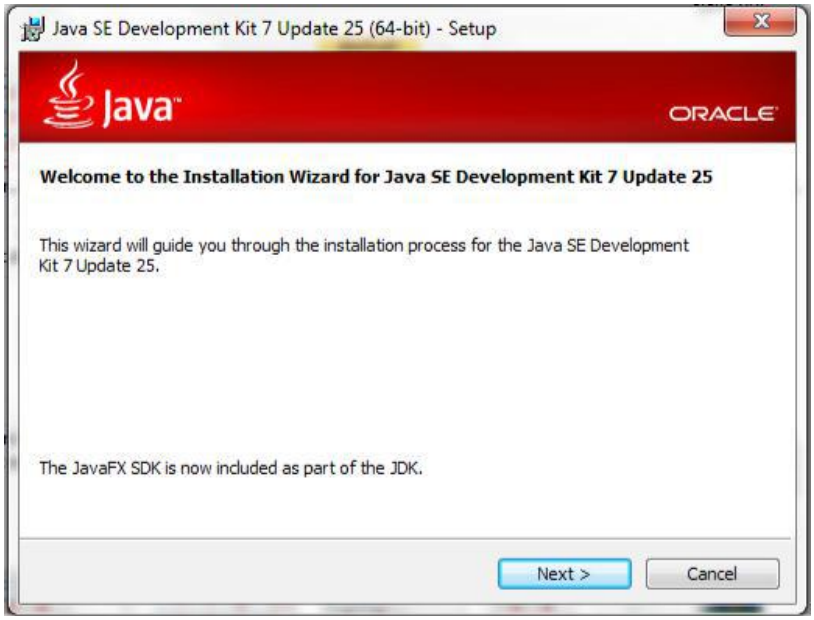
1. Habilite o download aceitando os termos da licença. **Accept License Agreement**
2. Escolha sua plataforma SO**: Windows** para plataforma **Windows,** por exemplo, e **click** em **Download**.

3. Selecione sua plataforma SO na coluna de Download, para iniciar o download

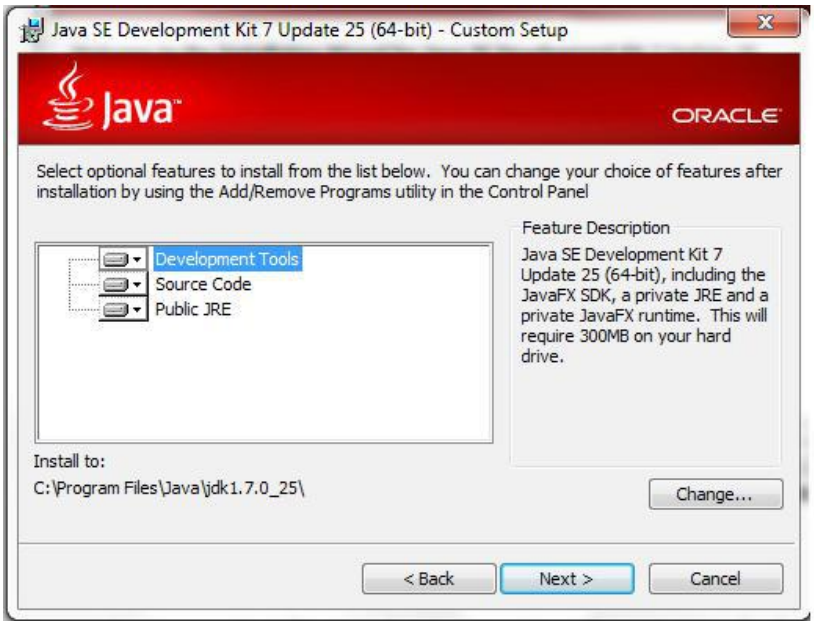
1.2 - Instalação do JDK em ambiente Windows

Para instalar o JDK no Windows, primeiro baixe-o no site da Oracle, é um simples arquivo executável que contém o Wizard de instalação.

1. Dê um clique duplo no arquivo **jdk-<versão>-windows-<arquitetura>.exe**, e espere até ele entrar no wizard de instalação.
2. Nesta tela Clique em **Next**.

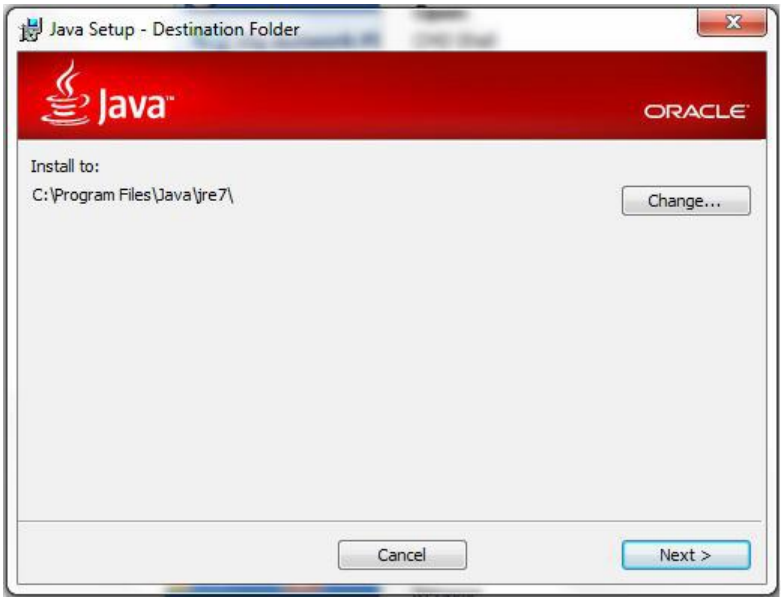


3. Agora devemos selecionar quais recursos instalaremos junto com o java (Ferramentas de desenvolvimento, o código fonte e o próprio java), e onde ele será instalado (marque esse caminho porque usaremos ele mais pra frente), deixe como está e clique em **Next**.



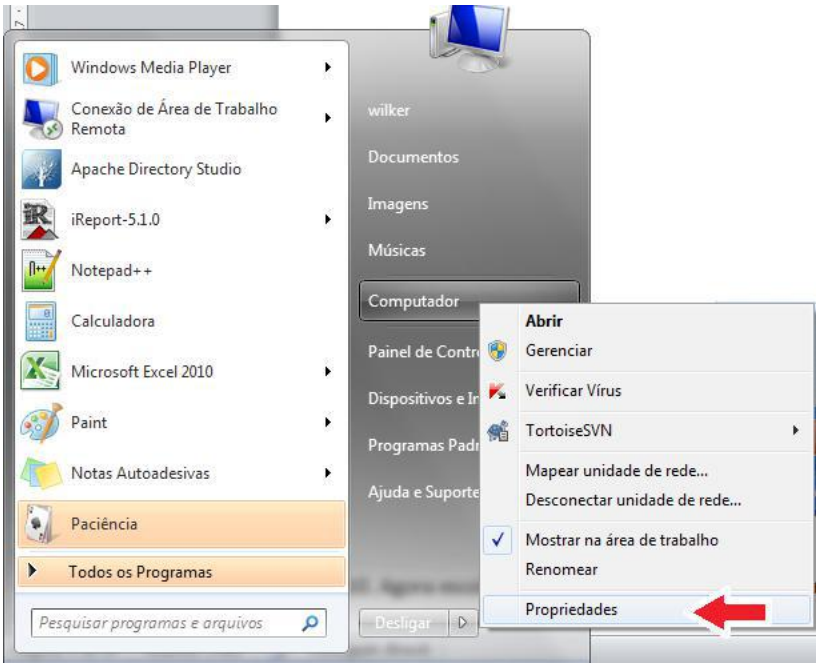
4. O processo de instalação iniciará.

5. Este passo só será executado caso você ainda não tenha instalado o JRE na sua máquina. Agora ele começará a instalar o **JRE(Java Runtime Environment)**. Assim como o **JDK**, ele também tem algumas opções. Deixe como está e clique em **Next**.

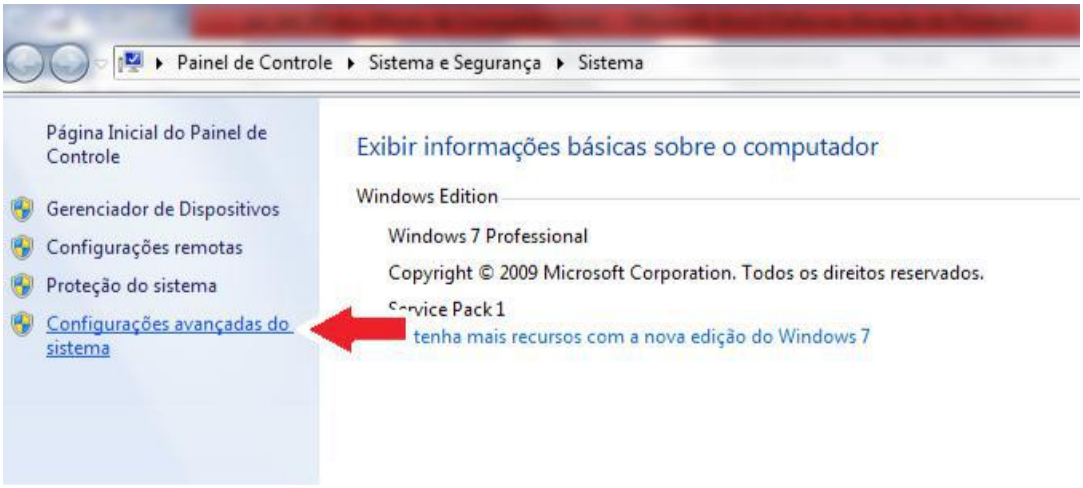


6. O processo de instalação da **JRE(Java Runtime Environment)** iniciará.
7. Agora seu **JDK** está instalado. Clique em **Finish**.

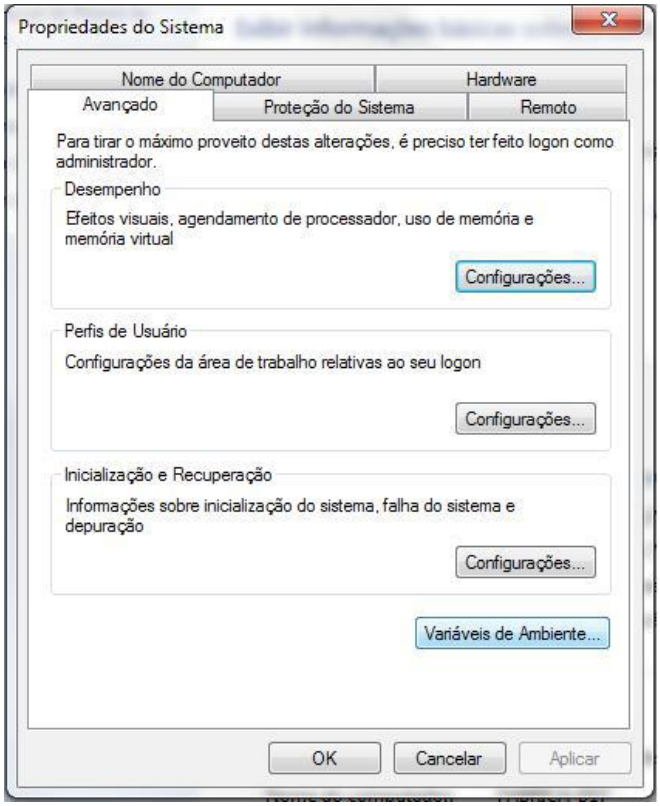
Agora vamos criar as ícone **Meu Computador** **Variáveis de Ambiente**. Clique com o botão direito em cima do e selecione a opção **Propriedades**.



9. Clique em **Configurações avançadas do sistema.**

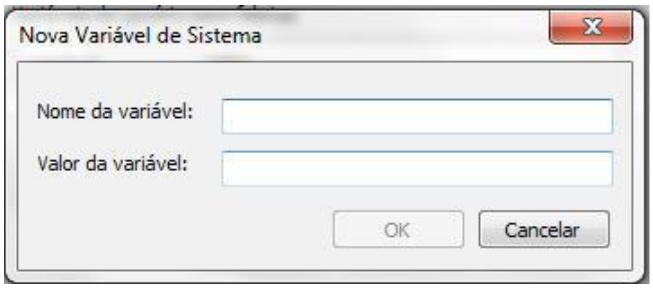


10. Agora escolha a aba **Avançado** e depois clique no botão **Variáveis de Ambiente.**

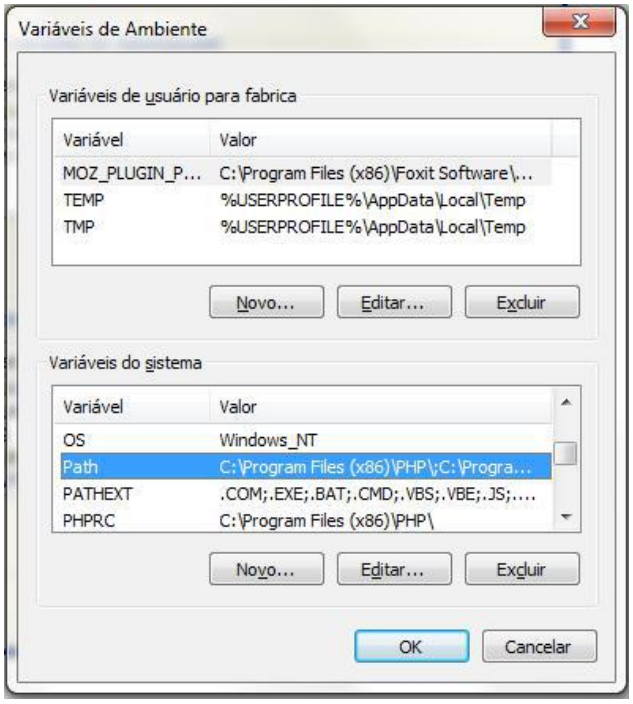


11. Nesta tela você verá na parte de cima, as variáveis de ambiente do usuário corrente, e embaixo, as variáveis de ambiente do computador, que serve para todos os usuários. Clique no botão **Novo** da parte de baixo.

12. Agora em **Nome da Variável** digite **JAVA_HOME** e em **Valor da Variável** digite o **Caminho** que você anotou no passo 3, ou seja, o caminho que foi instalado o JDK. Provavelmente vai estar em **C:\Program Files\Java\jdk1.7.0_25 ,** ou um caminho parecido com este dependendo da versão que foi instalado. Depois clique em **OK**.



13. Agora não vamos criar outra variável e sim alterar, para isso procure a variável **PATH**, ou **Path,** e clique no botão **Editar**.



14. Não altere o nome da variável e não apague os dados nela informados! Deixe como está e adicione no final do valor ***;%JAVA_ HOME%\bin,*** não esqueça do ponto-e-vírgula no início, assim você está adicionando mais um caminho à sua variável **Path**.
15. Agora abra o **prompt** e digite **javac -version** para mostrar a versão do **Java Compiler** e algumas opções, caso não apareça reveja os passos e confira se não esqueceu ou pulou nenhum deles e se o caminho colocado nas variáveis estão corretos.

1.3 - Instalação do JDK em ambiente Linux

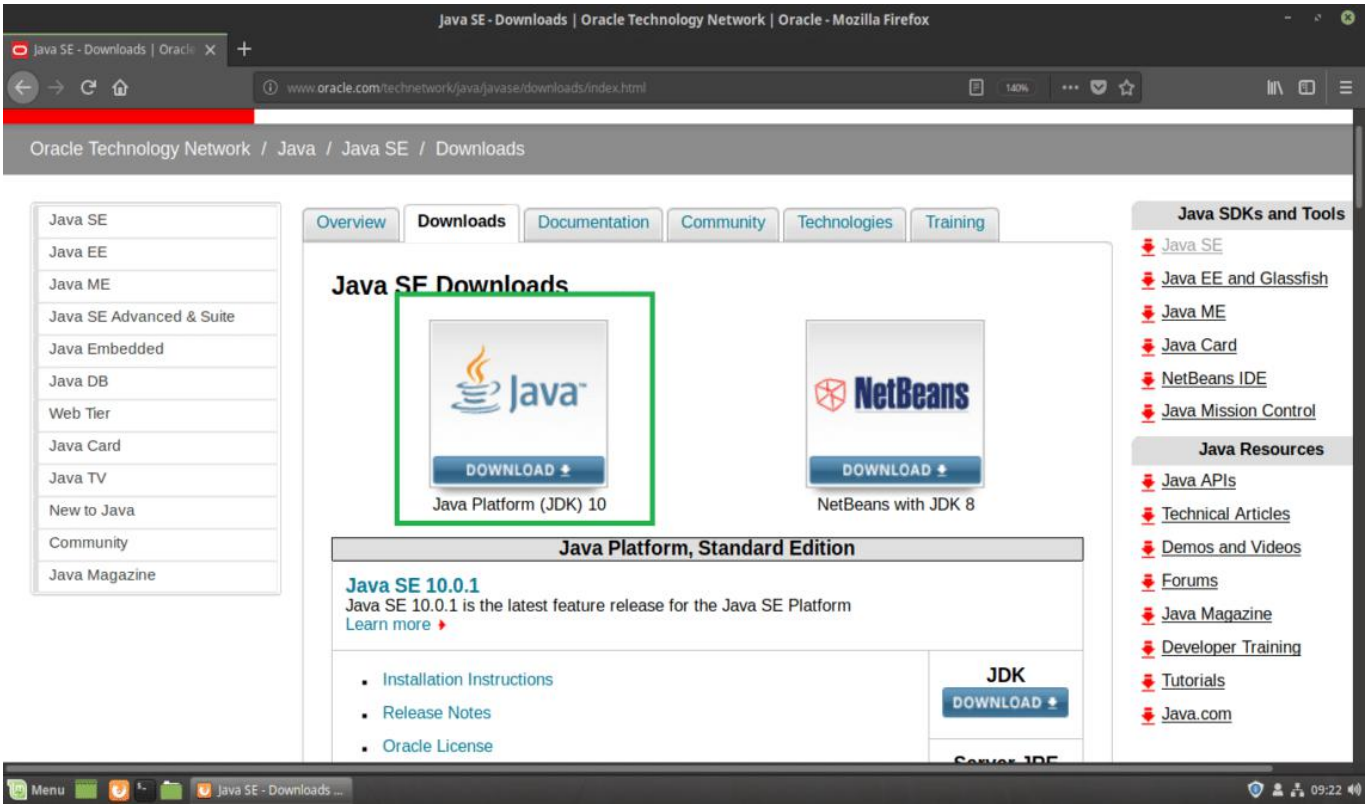
https://linuxhint.com/install_oracle_jdk_linux_mint_19/

Install Oracle Java Development Kit (JDK) 10 on Linux Mint 19

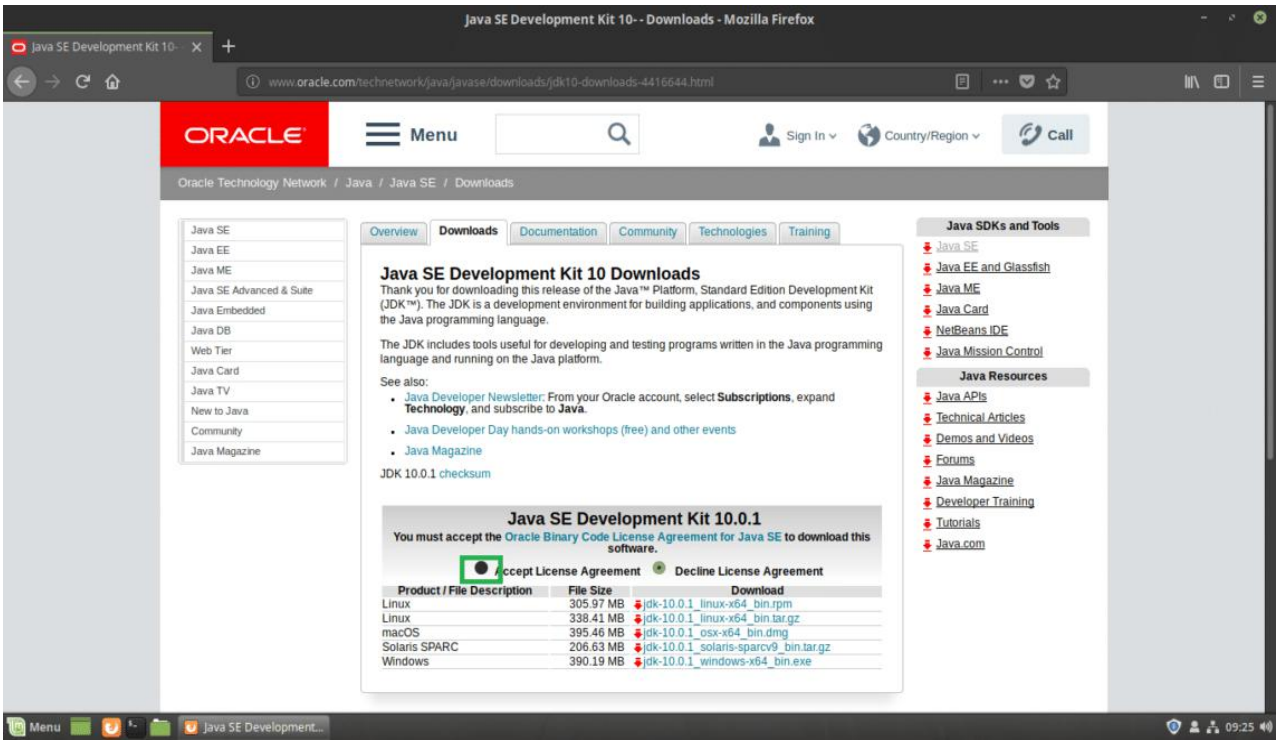
2 years ago
by Shahriar Shovon
In this article, I will show you how to install Oracle JDK 10 on Linux Mint 19. Let's get started.

Downloading Oracle JDK 10:

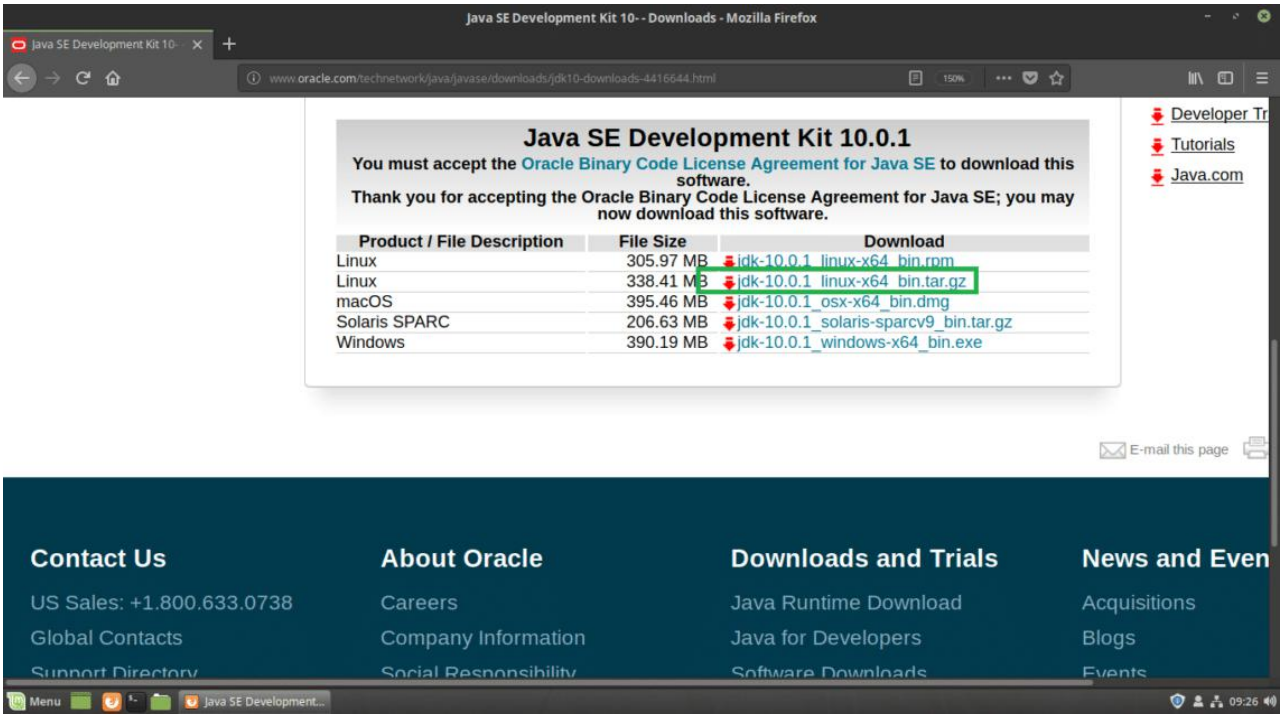
First go to the official website of Oracle Java Development Kit (JDK) at http://www.oracle.com/technetwork/java/javase/downloads/index.html and click on **Java Platform (JDK) 10** download link as marked in the screenshot below.



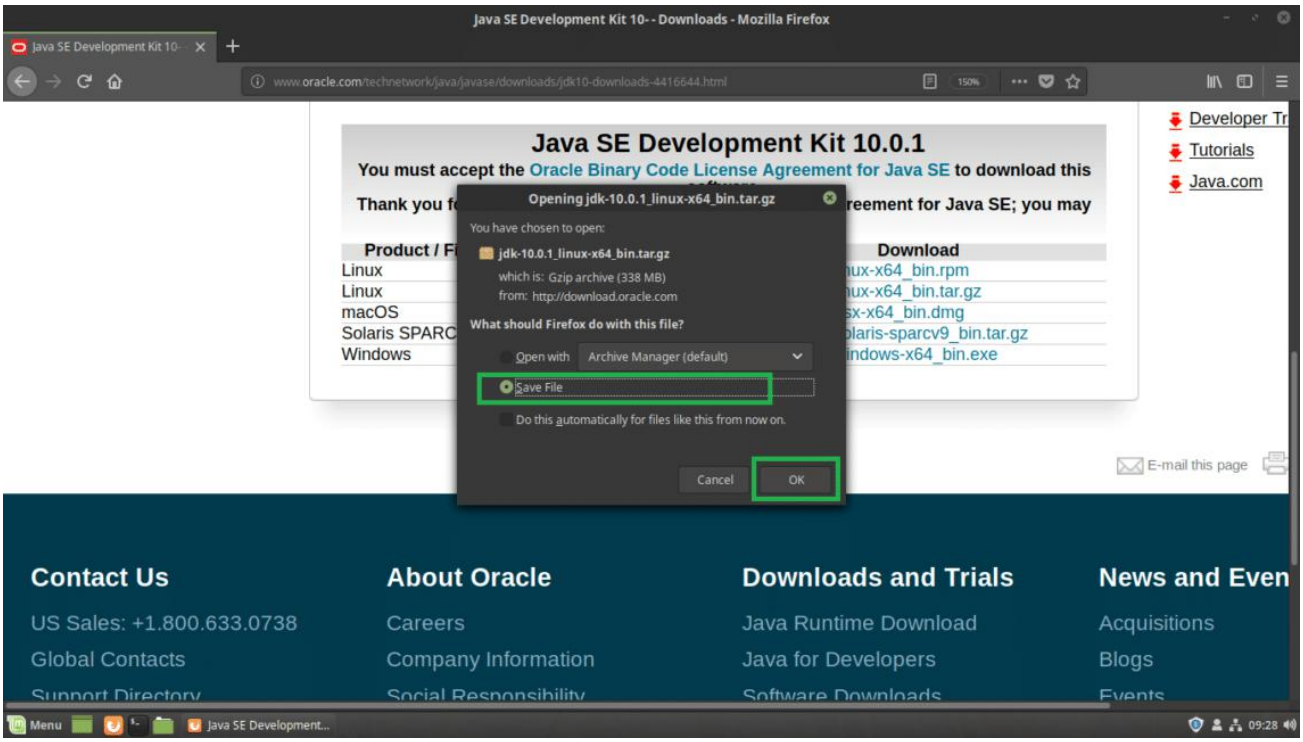
You should see the following page. Now click on the **Accept License Agreement** radio button as marked in the screenshot below.



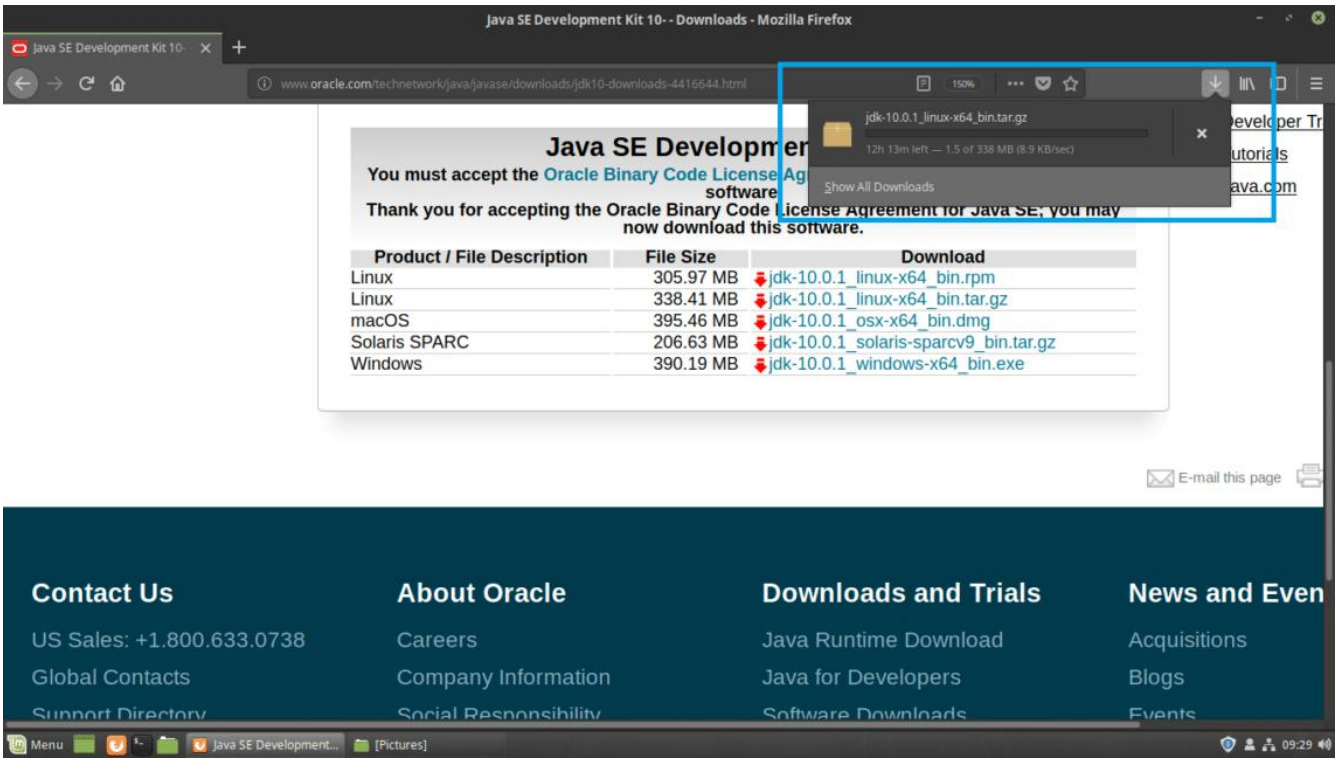
Once you accept the license agreement, you should be able to download JDK 10. Now click on the marked download link that ends in **.tar.gz** in the **Linux** section.



Your browser should prompt you to save the file. Click on **Save File** and then click on **OK**.

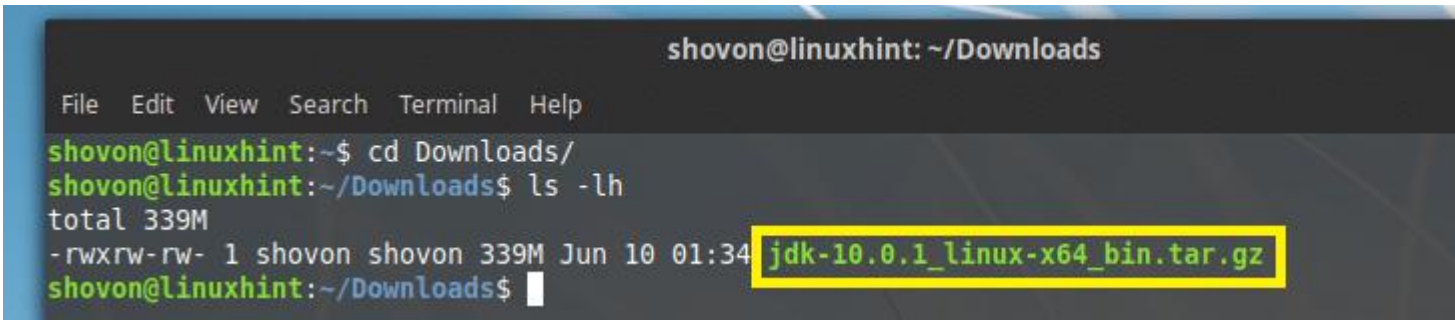


Your download should start.



Once the download is complete, you should find the **tar.gz** archive file in the **~/Downloads/** directory in your user's **HOME** directory. At the time of this writing, the name of the archive file is **jdk-10.0.1_linux-x64_bin.tar.gz**

$ cd ~/Downloads && ls -lh



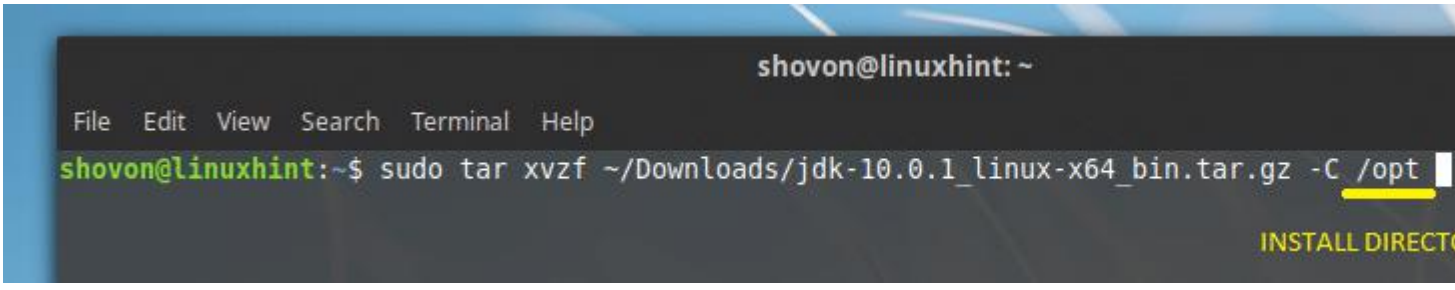
Installing Oracle JDK 10:

I am going to extract the Oracle JDK 10 **.tar.gz** archive in **/opt** directory. You may choose to install it somewhere else. It's up to you. Just adjust the rest of the article if you do install it somewhere other than **/opt**.

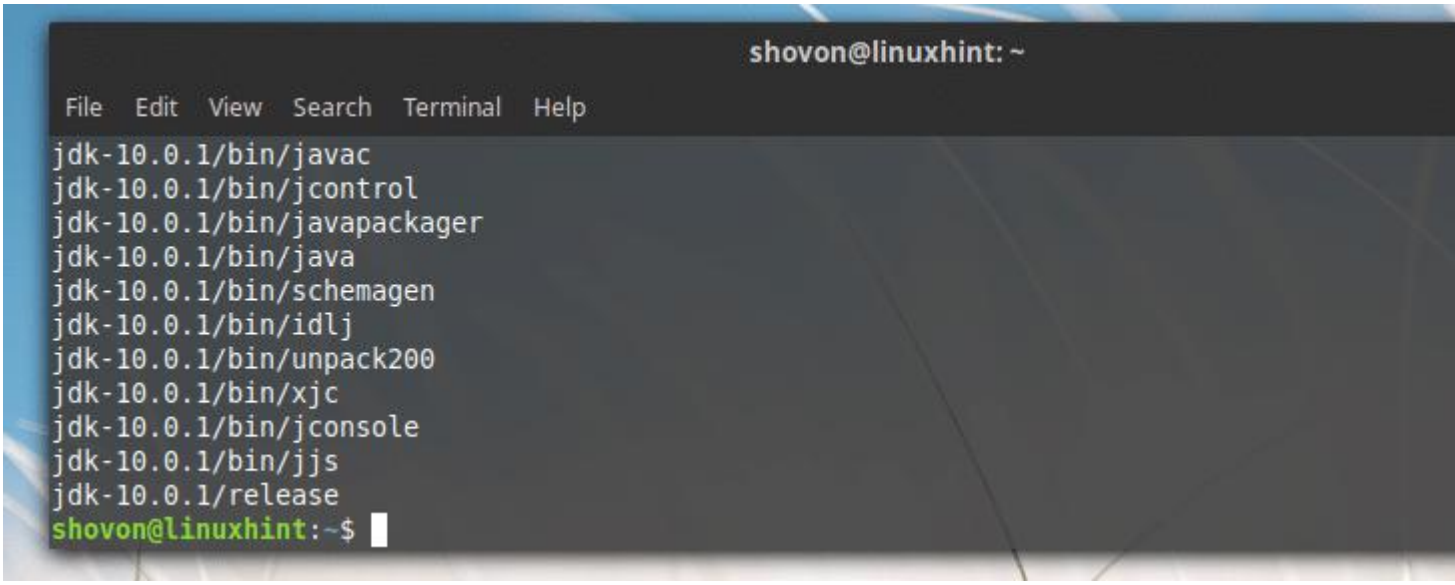
First extract the **Oracle JDK 10** archive file, **jdk-10.0.1_linux-x64_bin.tar.gz** to **/opt** directory with the following command:

$ sudo tar xvzf ~/Downloads/jdk-10.0.1_linux-x64_bin.tar.gz -C INSTALL_DIR

Note: In my case, **INSTALL_DIR** is **/opt**



The **.tar.gz** archive should be extracted to **/opt** directory.



Adding Oracle JDK 10 to the PATH:

Now that Oracle JDK 10 is installed in the **/opt** directory, a new sub directory should be created in **/opt**, which you can find out with the following command:

$ ls -lh /opt

As you can see from the marked section of the screenshot below, the directory name is **jdk-10.0.1**

Note the directory name, because you will need it to set up some environment variables.

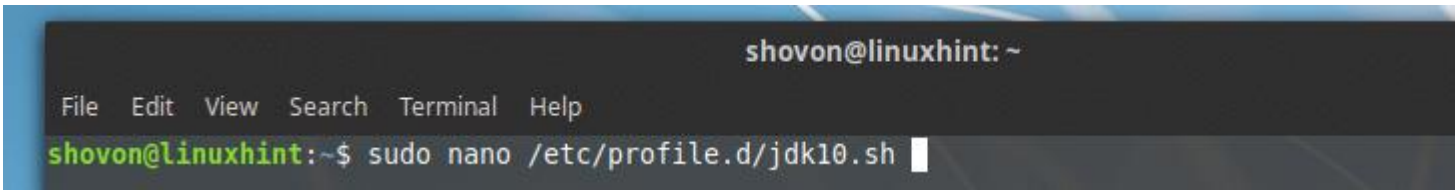
Note: The directory name may change as new updates of JDK 10 is released. Adjust it in the rest of the articles, if it changes by the time you read this article.

Now you need to create an environment variable **JAVA_HOME** with the value **/opt/jdk-10.0.1** and add **/opt/jdk-10.0.1/bin** to the **PATH** of your Linux Mint 19 operating system. So that you don't have to type in the full path when you run Java commands.

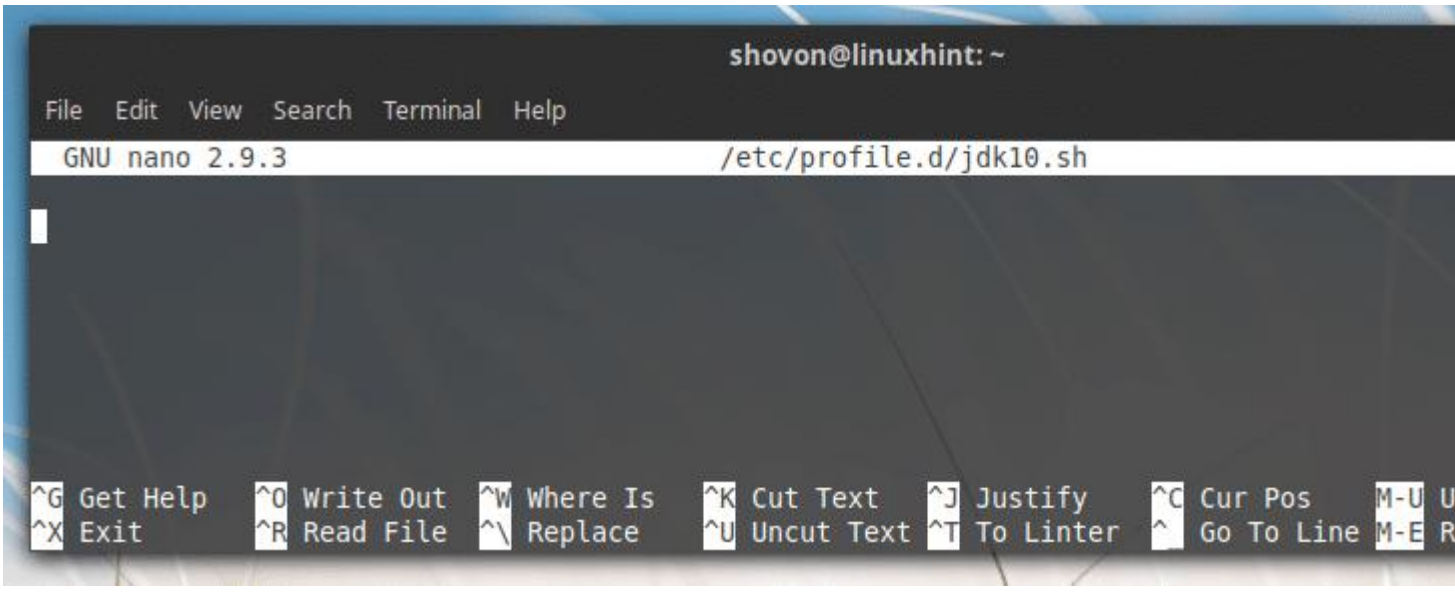
For example, when you want to compile a Java program, you don't have to run **/opt/jdk-10.0.1/bin/javac**, instead you can just run **javac** if you have the environment variables **JAVA_HOME** and **PATH** set up correctly. That's what I am going to do now.

First create a new file **jdk10.sh** in **/etc/profile.d** directory with the following command:

```
$ sudo nano /etc/profile.d/jdk10.sh
```



You should see the following window.

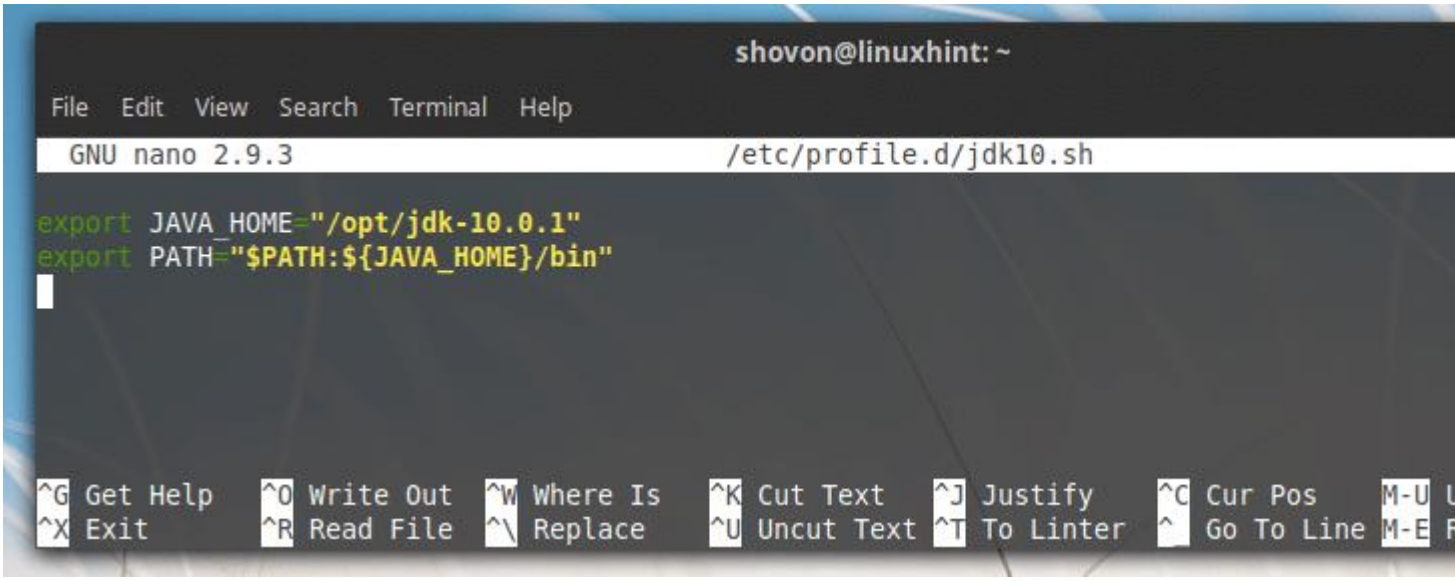


Now type in the following lines:

```
export JAVA_HOME="INSTALL_DIR/JDK_DIR"
export PATH="$PATH:${JAVA_HOME}/bin"
```

Note: It my case the **INSTALL_DIR** is **/opt** and the **JDK_DIR** is **jdk-10.0.1**, make sure to replace it correctly.

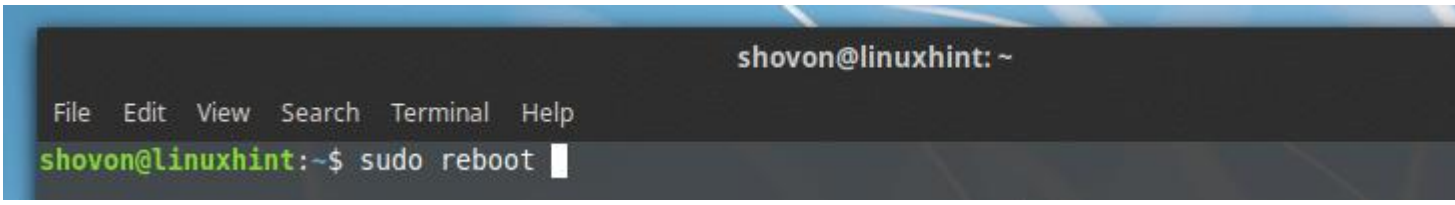
Finally, it should look something like this.



Now save the file by pressing **<Ctrl>** + **x** and then press **y** and then press **<Enter>**.

Now reboot your computer with the following command:

```
$ sudo reboot
```

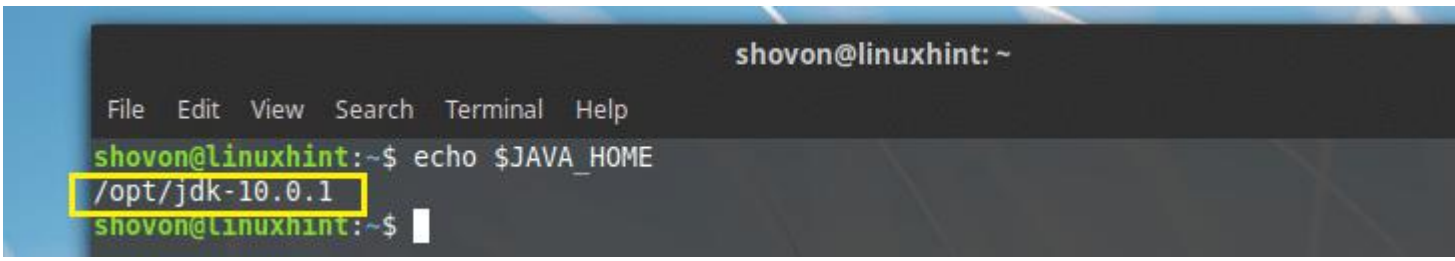


Once your computer starts, you can check whether the environment variables **JAVA_HOME** and **PATH** is set correctly.

To check whether **JAVA_HOME** is set, run the following command:

```
$ echo $JAVA_HOME
```

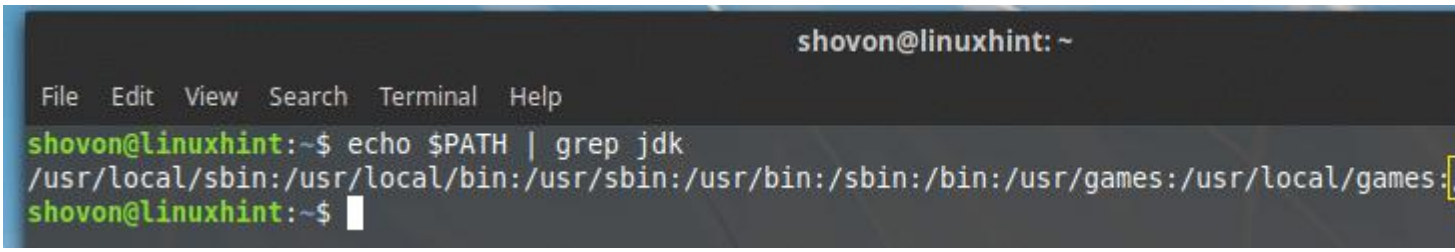
As you can see from the marked section of the screenshot below, the **JAVA_HOME** environment variable has the correct value set.



To check whether the **PATH** is correctly set, run the following command:

```
$ echo $PATH | grep jdk
```

As you can see from the marked section of the screenshot below, the JDK 10's binary directory is in the **PATH**.

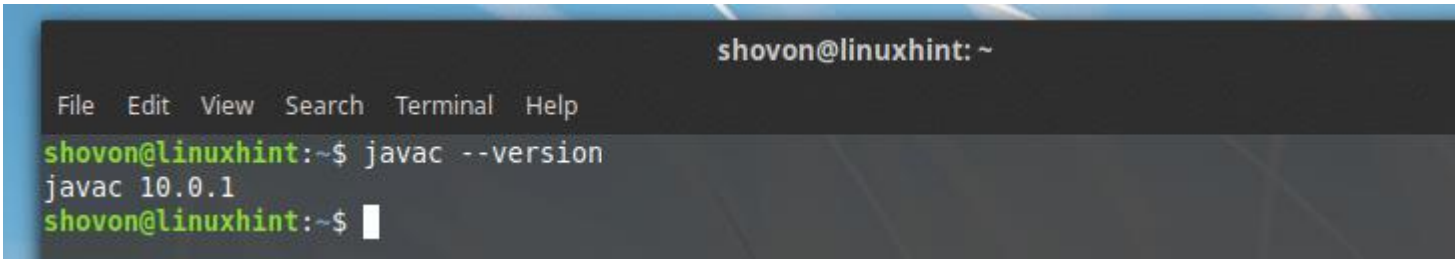


Testing Oracle JDK 10:

Now you can test whether JDK 10 is working correctly with the following command:

$ javac --version

As you can see from the output of the command, JDK 10.0.1 is working correctly.



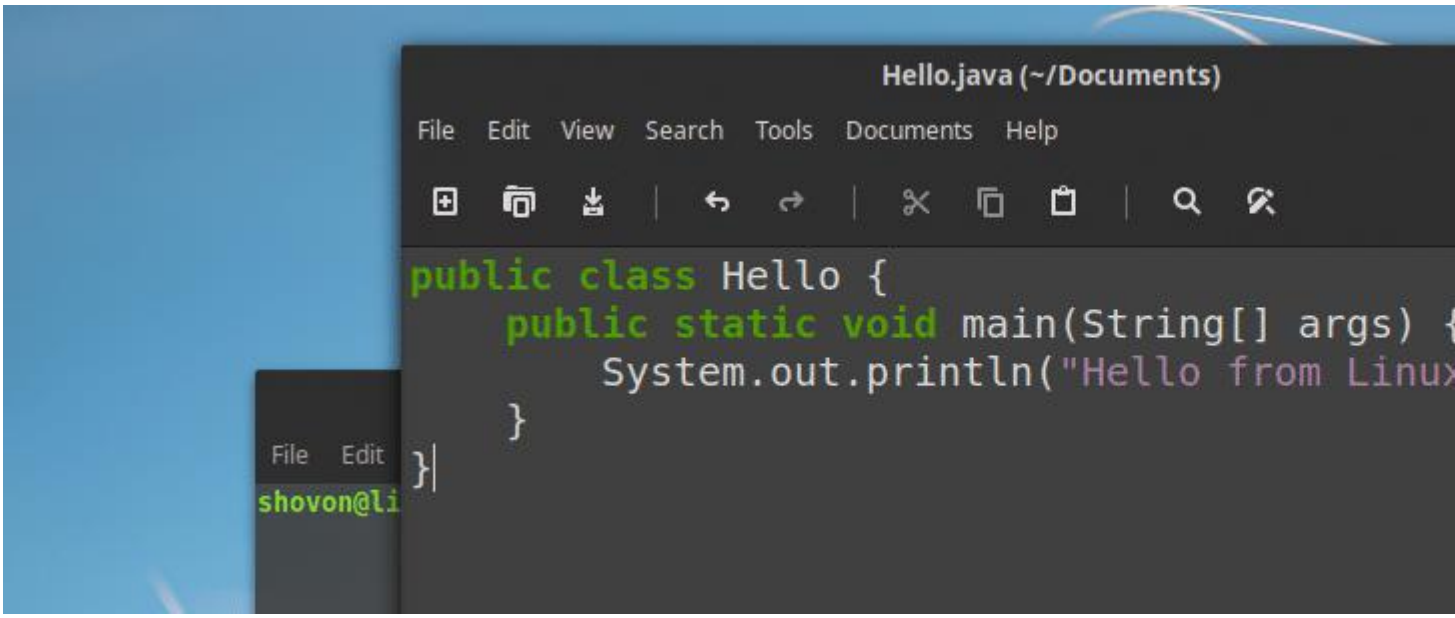
Compiling a Simple Java Program with Oracle JDK 10:

In this section I am going to write a simple Java program and compile it using Oracle JDK 10 on Linux Mint 19.

I am going to create a new file **Hello.java** in the **~/Documents** directory in my user's **HOME** directory. **Hello.java** is the Java source code file that I will compile with Oracle JDK 10.

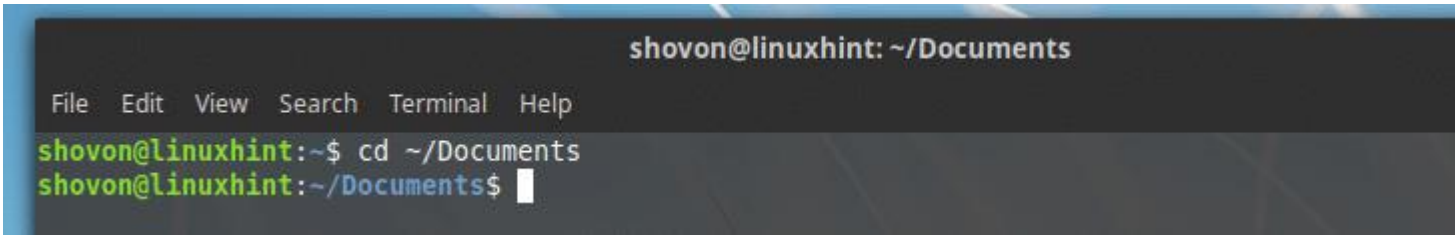
Here are the contents of **Hello.java**:

```
public class Hello {

public static void main(String[] args) {

System.out.println("Hello from LinuxHint!");

}

}
```



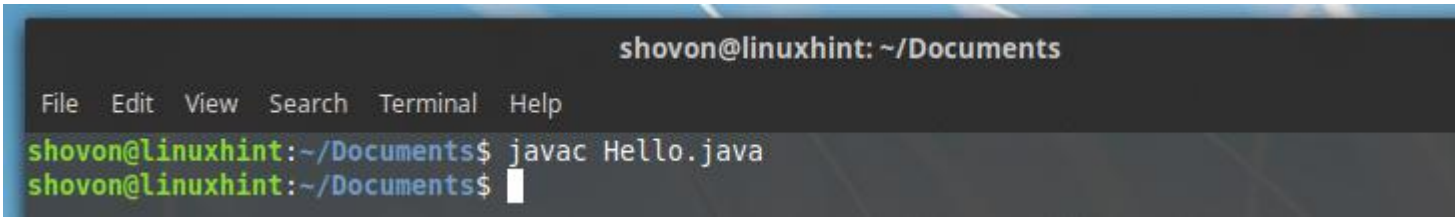
Before you can compile the Java program, go to the directory where **Hello.java** file is saved, which is in my case **~/Documents**

$ cd ~/Documents



Now run the following command to compile **Hello.java** Java source code file with Oracle JDK 10:

$ javac Hello.java



Running a Java Program with Oracle Java 10:

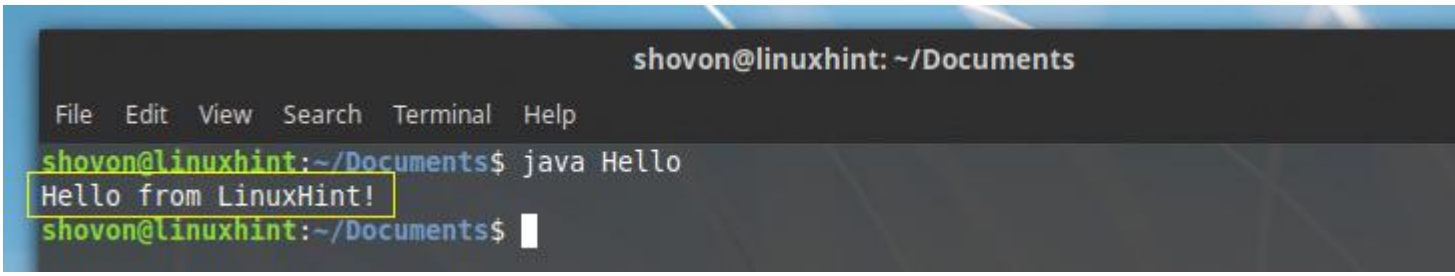
In this section, I will show you how to run the Java program that I compiled in the earlier section of this article.

To run the **Hello.java** Java source file I just compiled, from the **~/Documents** directory where the **Hello.java** source file is saved and compiled, run the following command:

$ java Hello

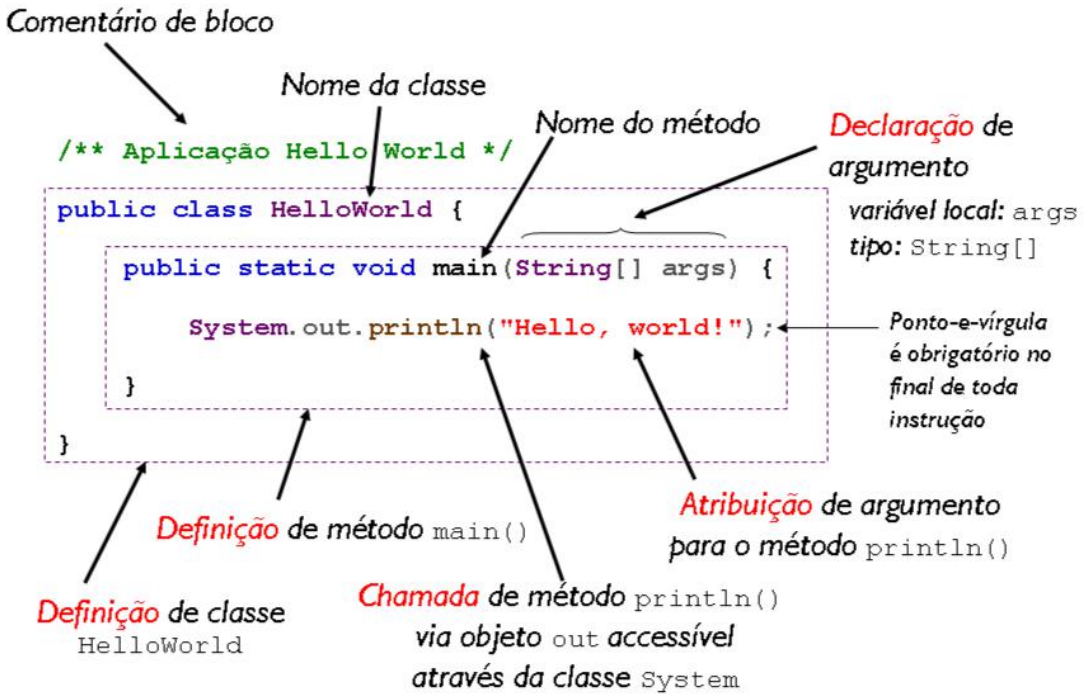
Note: When you run the java program, always leave the extension. For example, if the source file is **Hello.java**, it should be **Hello** when you want to run it.

As you can see from the marked section of the screenshot below, the correct output is displayed.



That’s how you install and use Oracle JDK 10 on Linux Mint 19. Thanks for reading this article.

Laboratório Java SE



3. Compile **HelloWorld.java** usando o compilador **javac**. O compilador **javac** foi instalado com seu **J2SE SDK**. Ele está no diretório **%JAVA_HOME%\bin** (Windows) ou **$JAVA_HOME/bin** (Linux). O resultado da compilação será a criação do arquivo **HelloWorld.class**.
> javac HelloWorld.java

Problema #1: se você receber a mensagem de erro abaixo

> HelloWorld.java

'javac' is not recognized as an internal or external command, operable program or batch file

Isto significa que **%JAVA_HOME%\bin** para Windows ou **$JAVA_HOME/bin** para Linux, não está configurada corretamente. Você pode tentar colocando **C:\Program Files\Java\jdk13\bin\javac HelloWorld.java** (para Windows) se quiser continuar sem configurar as variáveis de ambiente.

Solução: Sugiro que você reveja o passo 9 do item 1.2 (para windows) ou passo 7 do item 1.3 (para Linux), do exercício 1.

4. Certifique-se de que o arquivo **HelloWorld.class** foi criado. O arquivo **HelloWorld.class** contém a representação **bytecode** da classe **HelloWorld**.

> dir HelloWorld.class (para windows)

ou

- ***dir HelloWorld.class*** (para linux)

6. Rode o programa HelloWorld usando o comando **java**. O comando **java** inicia a **Máquina Vitual Java** e executa o programa **HelloWorld** deste exemplo. Um programa em java pode ter múltiplas classes e um conjunto de bibliotecas de classes . Neste exemplo, o programa HelloWorld possui uma única classe chamada **HelloWorld.class**. O comando **java** é o **Interpretador Java**.

>java HelloWorld

HelloWorld world

Problema #2: Se você receber a mensagem de erro

>java HelloWorld

Exception in thread "main" java.lang.UnsupportedClassVersionError: HelloWorld (Unsupported major.minor version 49.0)

É possível que você tenha compilado seu programa com uma versão diferente da que esteja tentando executá-lo.

Solução: certifique que sua variável de ambiente **PATH** esteja corretamente configurada para o diretório de instalação da versão apropriada do seu **JDK**.
Sugiro que você reveja o passo 9 do item 1.2 (para windows) ou passo 7 do item 1.3 (para Linux), do exercício 1.

Problema #3: Se você receber a mensagem de erro

>java HelloWorld

Exception in thread "main" java.lang.NoClassDefFoundError: HelloWorld

Provavelmente sua variável de ambiente **CLASSPATH** não está corretamente configurada com o diretório corrente.

Solução: tente usar o comando **java -classpath . HelloWorld** (Há um ponto **“.”** após o **“-classpath”**). Você verá mais sobre a classpath posteriormente.

>java -classpath . HelloWorld

7. Modifique **HelloWorld.java** como mostra a **Listagem-0.2** abaixo.

```
public class HelloWorld {

  /**
   * Comentário:
   * Primeiro programa em Java
   */
  public static void main( String[] args ){
    // também é um comentario
    // Imprime cadeia de caracteres "Hello world" na
    tela System.out.println("Olá Java");
    //modifique para imprimir mais
    System.out.println("Meu primeiro programa Java");
  }
}
```

Listagem-0.2: HelloWorld.java modificado

8. Compile e execute o programa. Observe a nova mensagem mostrada.

>javac

HelloWorld.java >java

HelloWorld

Olá Java

Meu primeiro programa Java

2.2 Usar o compilador Java, comando *javac*

Neste passo você aprenderá como usar algumas opções do compilador ***javac***.

1. Mostrar as opções do compilador **javac** usando a opção **–help**:

```
> javac -help
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
  -cp <path>                 Specify where to find user class files
  -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
  -d <directory>             Specify where to place generated class 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
  -X                         Print a synopsis of nonstandard options
  -J<flag>                   Pass <flag> directly to the runtime system
```

2. Execute o comando **javac** com a opção **-verbose**

```
>javac -verbose HelloWorld.java
```

3. Experiência com erros de compilação.

Modifique o programa **HelloWorld.java** como mostra a **Listagem-0-3** abaixo. A mudança causa um erro de sintaxe – a palavra-chave **static** está incorretamente digitada como **statict**.

```
public class HelloWorld {

 /**
  * Comentário:
  * Primeiro programa em Java
  */
 public statict void main( String[] args ){
   // também é um comentario
   // Imprime cadeia de caracteres na tela

}}System.out.println("Meu primeiro programa Java");
```

Listagem -0-3: Código contento erro de sintaxe

Compile HelloWorld.java com o comando javac.

Observe os erros

```
>javac HelloWorld.java
HelloWorld.java:6: <identifier>
expected
  public statict void main( String[] args ){
         ^
HelloWorld.java:12: ';' expected
}
^
2 errors
```

Remova o erro de sintaxe que você introduziu no código e recompile-o.

2.3 Usar o interpretador Java, comando *java*

Neste passo você aprenderá como usar algumas opções do interpretador **java**. Você pode pensar no comando como sua **Java Virtual Machine**.

1. Mostrar o uso da informações de ajuda do comando ***java*** com a opção ***-help*** ou ***-?*** .

```
>java -help
Usage: java [-options] class [args...]
     (to execute a class)
  or java [-options] -jar jarfile [args...]
     (to execute a jar file)

where options include:
  -client    to select the "client" VM
  -server    to select the "server" VM
```

-hotspot is a synonym for the "client" VM [deprecated]

The default VM is client.

-cp <class search path of directories and zip/jar files>
-classpath <class search path of directories and zip/jar
files> A ; separated list of directories, JAR archives,
and ZIP archives to search for class files.
-D<name>=<value>
set a system property
-verbose[:class|gc|jni]
enable verbose output
-version print product version and exit
-version:<value>
require the specified version to run
-showversion print product version and continue
-jre-restrict-search | -jre-no -restrict -search
include/exclude user private JREs in the version
search -? -help print this help message
-X print help on non- standard options
-ea[:<packagename>...|:<classname>]
-enableassertions[:<packagename>...|:<classname>]
enable assertions
-da[:<packagename>...|:<classname>]
-disableassertions[:<packagename>...|:<classname>]
disable assertions
-esa | -enablesystemassertions
enable system assertions
-dsa | -disablesystemassertions
disable system assertions
-agentlib:<libname>[=<options>]
load native agent library <libname>, e.g. -agentlib:hprof
see also, -agentlib:jdwp=help and -agentlib:hprof=help
-agentpath:<pathname>[=<options>]
load native agent library by full
pathname -javaagent:<jarpath>[=<options>]
load Java programming language agent, see java.lang.instrument

2. Execute o comando **java** com a opção **-verbose**

>java -verbose HelloWorld

3. Execute o comando **java** com a opção **-version**. Observe que o número da versão do JDK instalada é mostrado.

```
>java -version
java version "13"
Java(TM) SE Runtime Environment (build 13)
Java HotSpot(TM) 64-Bit Client VM (build 23.25-b01 , mixed mode)
```