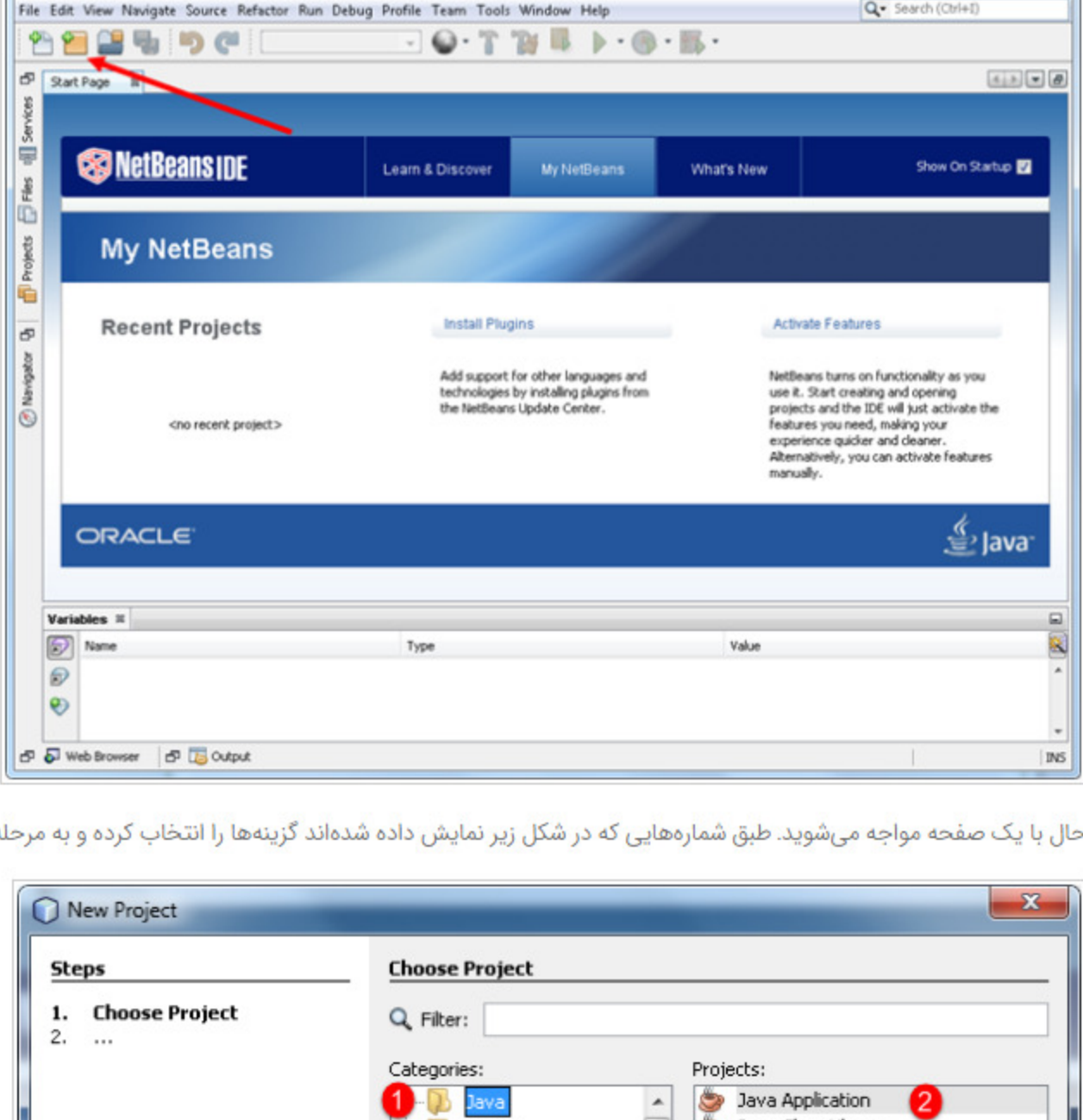
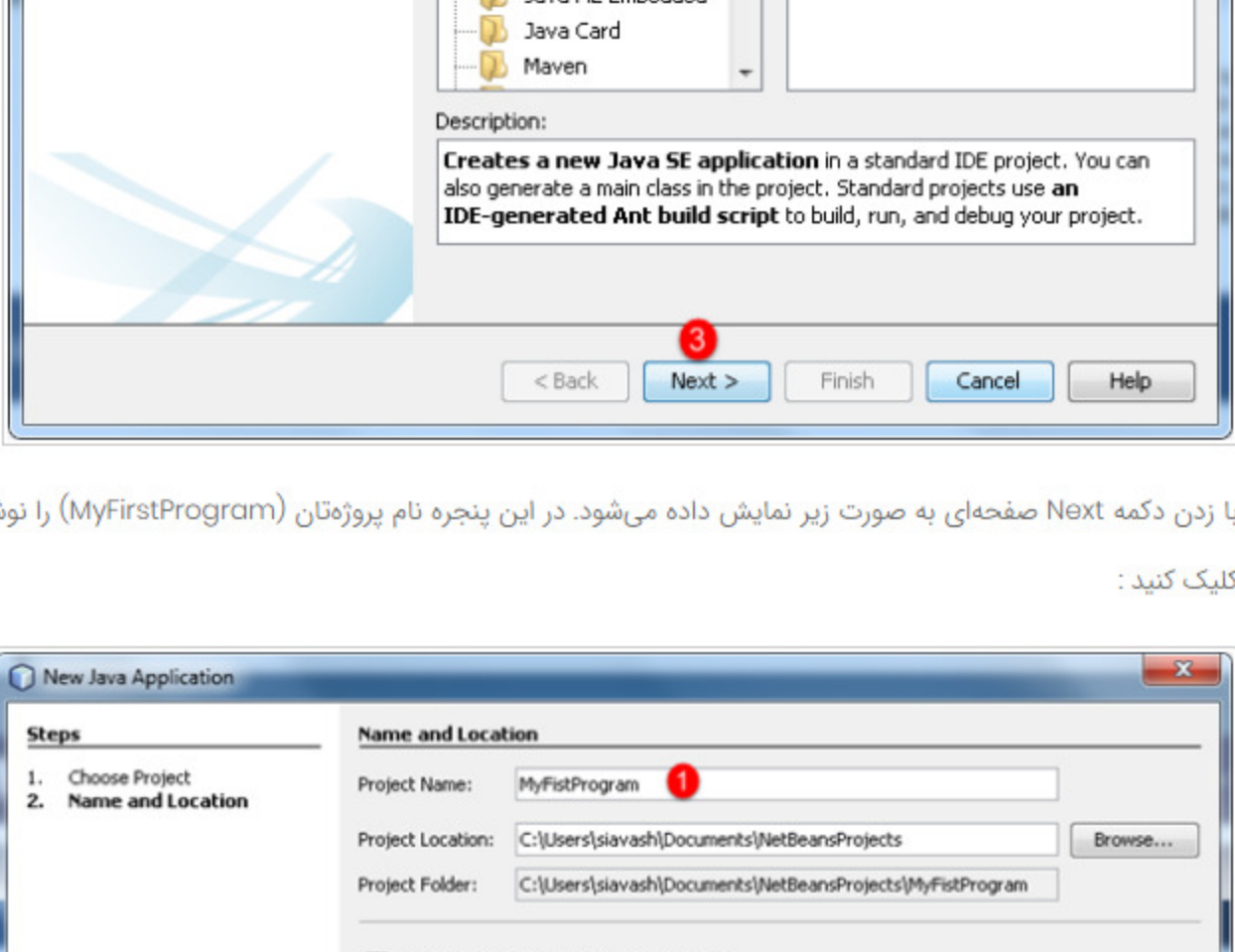


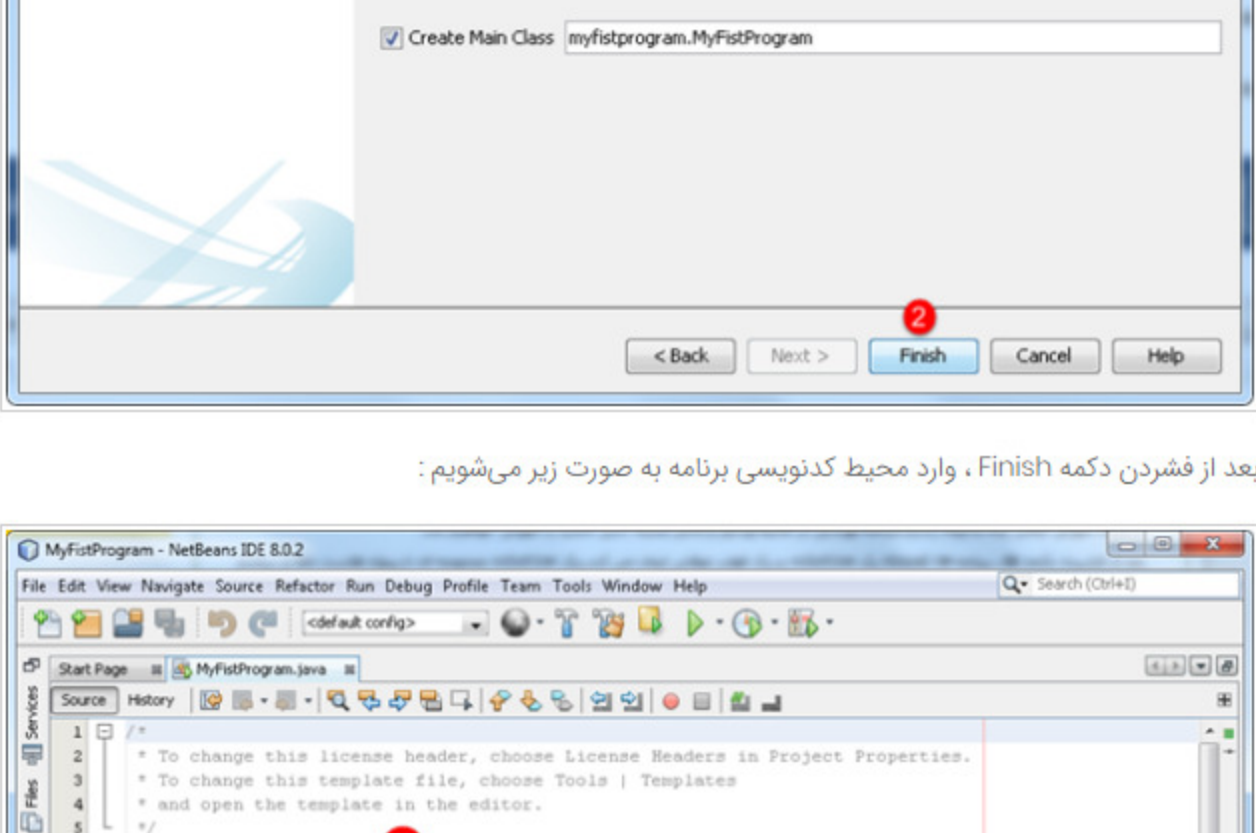
برنامه NetBeans را اجرا کنید. از مسیری که در شکل زیر نشان داده شده است یک پروژه جدید ایجاد کنید :



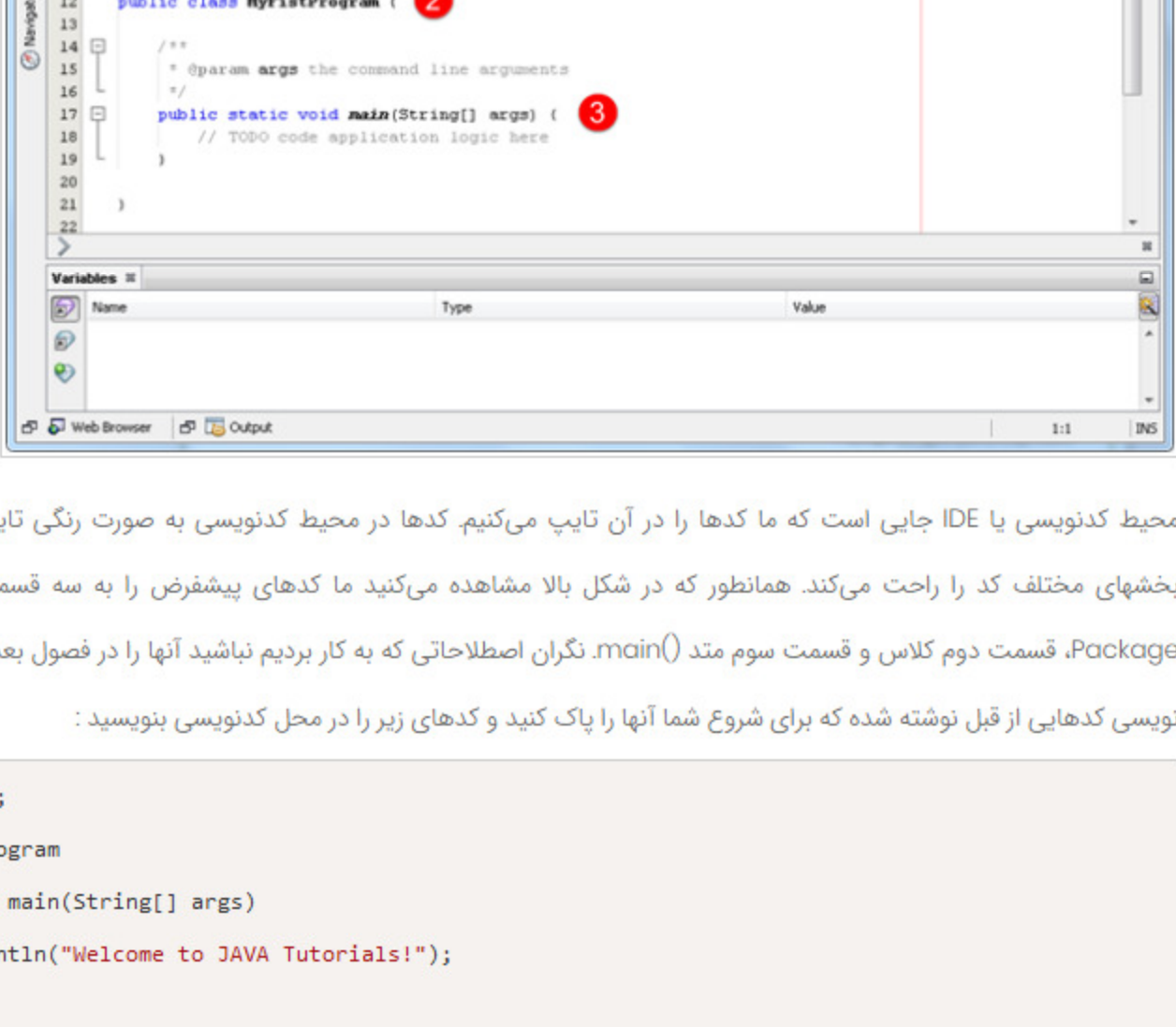
حال با یک صفحه مواجه می‌شوید. طبق شماره‌هایی که در شکل زیر نمایش داده شده‌اند گزینه‌ها را انتخاب کرده و به مرحله بعد بروید :



زدن دکمه Next صفحاتی به صورت زیر نمایش داده می‌شود. در این پنجره نام پروژه‌تان (MyFirstProgram) را نوشته و سپس بر روی دکمه Finish کلیک کنید :



بعد از فشردن دکمه Finish ، وارد محیط کدنویسی برنامه به صورت زیر می‌شویم :



محیط کدنویسی یا IDE جایی است که ما کدها را در آن تایپ می‌کنیم. کدها در محیط کدنویسی به صورت رنگی تایپ می‌شوند در نتیجه تشخیص بخشهای مختلف کد را راحت می‌کند. همانطور که در شکل بالا مشاهده می‌کنید ما کدهای پیشفرض را به سه قسمت تقسیم کرده‌ایم. قسمت اول Package، قسمت دوم کلاس و قسمت سوم متد main(). نگران اصطلاحاتی که به کار بردیم نباشید آنها را در فصول بعد توضیح خواهم داد. در محل کد نویسی کدهایی از قبل نوشته شده که برای شروع شما آنها را پاک کنید و کدهای زیر را در محل کدنویسی بنویسید :

```
package myfirstprogram;

public class MyFirstProgram
{
    public static void main(String[] args)
    {
        System.out.println("Welcome to JAVA Tutorials!");
    }
}
```

### ساختار یک برنامه در جاوا

مثال بالا ساده‌ترین برنامه‌ای است که شما می‌توانید در جاوا بنویسید. هدف در مثال بالا نمایش یک پیغام در صفحه نمایش است. هر زبان برنامه نویسی دارای قواعدی برای کدنویسی است. اجازه بدهید هر خط کد را در مثال بالا توضیح بدهیم. در خط اول Package تعریف شده است که شامل کدهای نوشته شده توسط شما است و از تداخل نامها جلوگیری می‌کند. درباره Package در درسهای آینده توضیح خواهیم داد.

در خط 4 آکولاد {} نوشته شده است. آکولاد برای تعریف یک بلوک کد به کار می‌رود. جاوا یک زبان ساخت یافته است که شامل کدهای زیاد و ساختارهای فراوانی می‌باشد. هر آکولاد باز {} در جاوا باید دارای یک آکولاد بسته {} نیز باشد. همه کدهای نوشته شده از خط 4 تا خط 9 یک بلوک کد است. در خط 3 یک کلاس تعریف شده است. درباره کلاسها در فصلهای آینده توضیح خواهیم داد. در مثال بالا کدهای شما باید در داخل یک کلاس نوشته شود. بدنه کلاس شامل کدهای نوشته شده از خط 4 تا 9 می‌باشد. خط 5 متد main() یا متد اصلی نامیده می‌شود. هر متد شامل یک سری کد است که وقتی اجرا می‌شوند که متد را صدا بزنیم. درباره متد و نحوه صدا زدن آن در فصول بعدی توضیح خواهیم داد. متد main() نقطه آغاز اجرای برنامه است. این بدان معناست که ابتدا تمام کدهای داخل متد main() و سپس بقیه کدها اجرا می‌شود. درباره متد main() در فصول بعدی توضیح خواهیم داد. متد main() و سایر متدها دارای آکولاد و کدهایی در داخل آنها می‌باشند و وقتی کدها اجرا می‌شوند که متدها را صدا بزنیم. هر خط کد در جاوا به یک سیمیگولن ; ختم می‌شود. اگر سیمیگولن در آخر خط فراموش شود برنامه با خطا مواجه می‌شود. مثالی از یک خط کد در جاوا به صورت زیر است :

```
System.out.println("Welcome to JAVA Tutorials!");
```

این خط کد پیغام Welcome to JAVA Tutorials! را در صفحه نمایش نشان می‌دهد. از متد println() برای چاپ یک رشته استفاده می‌شود. یک رشته گروهی از کاراکترها است که به وسیله دابل کوتیشن " محصور شده است. مانند : "Welcome to Visual C# Tutorials!". یک کاراکتر می‌تواند یک حرف، عدد، علامت یا ... باشد. در کل مثال بالا نحوه استفاده از متد println() نشان داده شده است. این متد یک کد از کلاس PrintStream بوده و از آن برای چاپ مقدر استفاده می‌شود. out یک فیلد استاتیک از کلاس System و کلاس System هم یک کلاس از پیش تعریف شده در جاوا می‌باشد. جاوا فضای خالی و خطوط جدید را نادیده می‌گیرد. بنابراین شما می‌توانید همه برنامه را در یک خط بنویسید. اما اینکار خواندن و اشکال زدایی برنامه را مشکل می‌کند. یکی از خطاهای معمول در برنامه نویسی فراموش کردن سیمیگولن در پایان هر خط کد است. به مثال زیر توجه کنید :

```
System.out.println("Welcome to JAVA Tutorials!");
```

جاوا فضای خالی بالا را نادیده می‌گیرد و از کد بالا اشکال نمی‌گیرد. اما از کد زیر ایراد می‌گیرد.

```
System.out.println(
    "Welcome to JAVA Tutorials!");
```

به سیمیگولن آخر خط اول توجه کنید. برنامه با خطای نحوی مواجه می‌شود چون دو خط کد مربوط به یک برنامه هستند و شما فقط باید یک سیمیگولن در آخر آن قرار دهید. همیشه به یاد داشته باشید که جاوا به بزرگی و کوچکی حروف حساس است. یعنی به طور مثال MAN و man در جاوا با هم فرق دارند. رشته‌ها و توضیحات از این قاعده مستثنی هستند که در درسهای آینده توضیح خواهیم داد. مثلاً کدهای زیر با خطا مواجه می‌شوند و اجرا نمی‌شوند :

```
system.out.println("Welcome to JAVA Tutorials!");
SYSTEM.OUT.PRINTLN("Welcome to JAVA Tutorials!");
sYsTem.oUt.pRinTln("Welcome to JAVA Tutorials!");
```

تغییر در بزرگی و کوچکی حروف از اجرای کدها جلوگیری می‌کند. اما از کد زیر کاملاً بدون خطا است :

```
System.out.println("Welcome to JAVA Tutorials!");
```

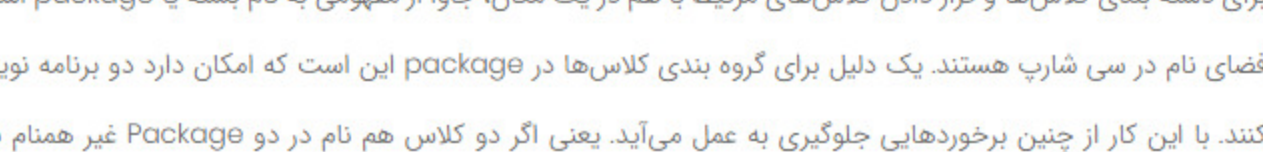
همیشه کدهای خود را در داخل آکولاد بنویسید.

```
{
    statement1;
}
```

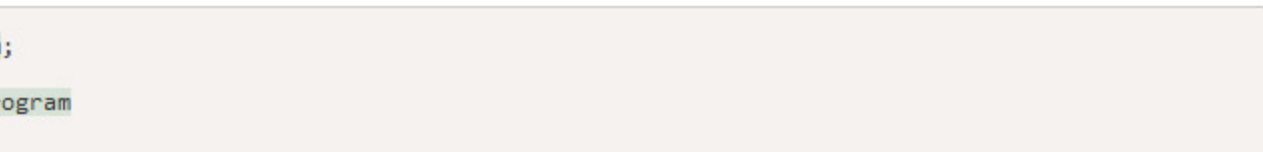
این کار باعث می‌شود که کدنویسی شما بهتر به چشم بیاید و تشخیص خطاها راحت‌تر باشد. یکی از ویژگیهای مهم جاوا نشان دادن کدها به صورت تو رفتگی است بدین معنی که کدها را به صورت تو رفتگی از هم تفکیک می‌کند و این در خوانایی برنامه بسیار مؤثر است.

### ذخیره پروژه و اجرای برنامه

برای ذخیره پروژه و برنامه می‌توانید به مسیر File > Save All بروید یا از کلیدهای میانبر Ctrl+Shift+S استفاده کنید. همچنین می‌توانید از قسمت Toolbar بر روی شکل زیر کلیک کنید :



و برای اجرای برنامه هم از فلش سبز رنگ موجود در Toolbar و یا دکمه F6 استفاده کنید :



با اجرای برنامه بالا مشاهده می‌کنید که رشته Welcome to JAVA Tutorials! در خروجی برنامه به صورت زیر نمایش داده می‌شود :



وجود خط سبز در پایین فلش قرمز در شکل بالا نشان دهنده اجرای بدون نقص برنامه می‌باشد. حال که با خصوصیات و ساختار اولیه جاوا آشنا شدید در دسهای آینده مطالب بیشتری از این زبان برنامه نویسی قدرتمند خواهید آموخت.

### استفاده از Package

برای دسته بندی کلاس‌ها و قرار دادن کلاس‌های مرتبط با هم در یک مکان، جاوا از مفهومی به نام بسته یا package استفاده می‌کند. Package معادل فضای نام در سی شارپ هستند. یک دلیل برای گروه بندی کلاس‌ها در package این است که امکان دارد دو برنامه نویس از دو کلاس هم نام استفاده کنند. با این کار از چنین برخوردهایی جلوگیری به عمل می‌آید. یعنی اگر دو کلاس هم نام در دو Package غیر همان باشند مشکلی به وجود نمی‌آید. همانطور که در مثال بالا دیدید به طور پیشفرض NetBeans هنگام ایجاد برنامه یک Package همانما با اسمی که برای برنامه انتخاب کرده‌ایم با حروف کوچک و در داخل این Package هم کلاسی به همین اسم ایجاد می‌کند :

```
package myfirstprogram;

public class MyFirstProgram
{
    ...
}
```

برای وارد کردن کلاس یک Package در داخل Package دیگر از کلمه کلیدی import به صورت زیر استفاده می‌شود :

```
import PackageName.ClassName
```

همانطور که در مثال بالا مشاهده می‌کنید برای استفاده از کلاسی که در یک Package قرار دارد در Package دیگر ابتدا کلمه import سپس نام Package ، بعد علامت نقطه و در آخر نام کلاس را می‌نویسیم. مثلاً برای استفاده از کلاس MyFirstProgram مربوط به پکیج myfirstprogram به صورت زیر عمل می‌شود :

```
import myfirstprogram.MyFirstProgram;
```

بسته‌ها را می‌توان به صورت تو در تو تعریف کرد. در این حالت در تعریف بسته یک کلاس، از بیرونی‌ترین بسته شروع کرده و هر بسته را با نقطه (.) به بسته بعدی متصل می‌کنیم :

```
import firstPackage.secondPackage.ClassName
```

نکته‌ای که بهتر است در همین جا به آن اشاره کنم این است که اگر بخواهید کد زیر را بدون استفاده از NetBeans کامپایل و اجرا کنید، کامپایل می‌شود ولی در زمان اجرا با خطای Error: Could not find or load main class MyFirstProgram مواجه می‌شوید :

```
package myfirstprogram;

public class MyFirstProgram
{
    public static void main(String[] args)
    {
        System.out.println("Welcome to JAVA Tutorials!");
    }
}
```

کد بالا را با نام و پسوند MyFirstProgram.java در درایو D ذخیره کنید. حال cmd را اجرا کرده و سعی کنید کد بالا را کامپایل و اجرا نمایید :

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\YounesJava>cd:
D:\>javac MyFirstProgram.java

D:\>java MyFirstProgram
Error: Could not find or load main class MyFirstProgram

D:\>java -cp . myfirstprogram.MyFirstProgram
Welcome to JAVA Tutorials!
```

همانطور که مشاهده می‌کنید برای کامپایل کد بالا به صورت زیر عمل کرده‌ایم :

```
javac MyFirstProgram.java
```

اما برای اجرای کد، اگر از دستور زیر استفاده کنیم با خطا مواجه می‌شویم :

```
java MyFirstProgram
```

دلیل وجود خطا، خط اول کد، یعنی package myfirstprogram; می‌باشد. برای رفع این مشکل یا باید خط مذکور را از برنامه حذف و برنامه را مجدداً کامپایل و اجرا کرد یا اگر آن را پاک نکنید باید از کد زیر برای اجرای کد استفاده کنید :

```
java -cp . myfirstprogram.MyFirstProgram
```

کد به این صورت برنامه اجرا و پیغام Welcome to JAVA Tutorials چاپ می‌شود. در درس‌های آینده توصیه می‌کنیم که اگر می‌خواهید از NotePad و

برای اجرای کدها استفاده کنید این خط را از اول هم قایل‌ها حذف و سپس برنامه را کامپایل و اجرا کنید.