

به نام خدا

آموزش دوم

حال نوبت به ساخت اولین پروژه خود در جاوا می رسد که امید داریم با طی کردن گام ها یکی پس از دیگر همانطور که توضیح داده می شود مشکلی در اجرای اولین برنامه جاوا خود نداشته باشید.

پیش از آغاز آموزش نیاز است با مفهوم واژه Compile در محیط برنامه نویسی آشنا شویم چرا که در طول آموزش ها از این واژه به کرات استفاده خواهد شد. معنوی لغوی واژه فوق الذکر در زبان فارسی جمع آوری کردن می باشد. به طور مثال کسی که برای تالیف یک کتاب از منابع مختلف استفاده می کند و از جاهای گوناگون مطالب کتابش را تهیه می کند در واقع Compile می کند. در محیط برنامه نویسی این واژه به زمانی اطلاق می شود که ما به عنوان برنامه نویس تعدادی Statement یا دستور نوشته ایم و می خواهیم آن ها را در کنار یکدیگر جمع نماییم. به عبارت دیگر در حین Compile شدن، دستوراتی که ما به عنوان برنامه نویس نوشته ایم به دستورات قابل فهم برای کامپیوتر که همان صفر و یک می باشند تبدیل خواهند شد و در نهایت برنامه شما اجرا می گردد. حال از این پس زمانیکه در طول آموزش به واژه Compile برخورد کردیم منظور همان مراحلی است که پیش از اجرای برنامه نوشته شده توسط اکلیپس صورت می گیرند تا کلیه عناصر برنامه ما آماده اجرا شوند.

به منظور ایجاد یک پروژه جدید در محیط برنامه نویسی اکلیپس از منوی File گزینه New سپس Java Project را انتخاب نمایید. در بخش Project Name به دلخواه می توانید نامی را برای پروژه خود در نظر بگیرید. در این آموزش نام پیشنهادی Hello

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

World می باشد. حال در بخش JRE نسخه مد نظر برای اجرای کدهای جاوای خود را مشخص ساخته و گزینه Use and execution environment JRE را انتخاب نمایید. در بخش Project Layout گزینه Create separate folders for sources and class files را انتخاب نمایید. حال مابقی گزینه ها را در حالت Default یا پیش فرض گذاشته و تغییر در آن ها ایجاد نکرده و گزینه Finish را بزنید (اگر به خاطر داشته باشید در مقدمه آموزش ها نحوه نصب JDK و فعال سازی آن در محیط برنامه نویسی جاوا آموزش داده شد. در واقع شما با نصب JDK به JRE یا Java Runtime Environment هم که برای برنامه نویسی جاوا ضروری است دسترسی خواهید داشت).

تا اینجا شما اولین گام برای ساخت یک پروژه جاوا را برداشته اید. شما در محیط اکتیپس در بخش Package Explorer فولدر پروژه ای تحت عنوان Hello World خواهید دید. حال با کلیک بر روی نام پروژه ای که جدیداً ساخته اید آن را انتخاب نموده سپس کلیک راست نمایید. با انتخاب گزینه New پنجره دیگری باز خواهد شد و از میان گزینه های موجود روی گزینه Class کلیک نمایید. پنجره ای تحت عنوان New Java Class باز خواهد شد. به منظور دسترسی به چنین پنجره ای می توانید از منوی اصلی گزینه File را انتخاب نموده سپس روی گزینه New کلیک نمایید و در نهایت گزینه Class را انتخاب نمایید. حال در پنجره Class در بخش Name نامی را برای Class مد نظر انتخاب نمایید. در اینجا ما نام FirstProgram را انتخاب می کنیم. این نام می بایست با نام فایلی به پسوند java. که در این مرحله ایجاد می گردد یکسان باشد که اکتیپس این کار را به صورت خود کار برای ما انجام خواهد داد. چنانچه نام Class با نام فایل با پسوند java. یکی نباشد بسته به اینکه این Class ایجاد شده در کجای برنامه مورد استفاده قرار گیرد منجر به Error در حین Compile شدن برنامه خواهد

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

شد(چنانچه نامی به غیر از آنچه می بایست انتخاب نمایید انتخاب کرده اید، می توانید روی فایل مد نظر کلیک راست نموده و از بخش Refactor گزینه Rename را انتخاب نمایید). لازم به ذکر است حرف اول نامی که برای Class خود انتخاب می نمایید می بایست با حرف بزرگ شروع شود. در حقیقت برای نامگذاری یک Class شما نمی توانید از - و فضای خالی استفاده کنید. تنها علامت نگارشی که در نامگذاری یک کلاس می توان استفاده کرد _ می باشد. حال گزینه public static void main(String[] args) را بزنید و دیگر گزینه ها را در حالت پیش فرض قرار داده و دکمه Finish را بزنید. اکنون درون کلاس HelloWorld شما می بایست کدهایی به شکل زیر را مشاهده نمایید:

```

1 class FirstProgram {
2     /**
3      * @param args
4      */
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7     }
8 }

```

خطی از برنامه که در برگیرنده public static void main(String[] args) می باشد به جاوا می گوید که این نقطه به منزله نقطه آغازین برنامه است. در حقیقت واژه main به ما می گوید که این بخش از برنامه جایی است که برنامه از آن آغاز می گردد. همانطور که ما برای نگارش، از پرانتز برای تبدیل تعدادی واژه به یک گروه استفاده می کنیم مثل (جاوا، زبان برنامه نویسی و اکیپس)، در زبان برنامه نویسی جاوا نیز به منظور نشان دادن عناصری که به یک گروه تعلق دارند از { و } استفاده می کنیم. به عبارت دیگر هرآنچه که مابین { آغازین و } پایانی باشد از یک گروه می باشد. به این گروه ها در زبان برنامه نویسی جاوا Block گفته می شود. در کد بالا { اول بعد از واژه Firstprogram و آخرین } نشاگر یک block می باشند که کل برنامه جاوای ما را

در بر می گیرند. نکته ای که اینجا می بایست مد نظر قرار داده شود این است که block ها را می توان درون دیگر block ها قرار داد. در خط ۵ و خط ۷ یک جفت {} دیگر مشاهده می شود که دربرگیرنده یک Method با نام main هستند (در بخش های آتی به توضیح اینکه یک Method چیست خواهیم پرداخت). آنچه در این block قرار می گیرد به منزله نقطه آغازین برنامه خواهد بود. در خط ششم علامت // دیده می شود که اصطلاحاً در زبان برنامه نویسی جاوا Comment گفته می شود. در حقیقت Comment ها در زبان برنامه نویسی صرفاً برای خود برنامه نویس هستند و از چشمان Compiler مخفی می مانند. به عبارت دیگر چنانچه شما بخواهید نکته ای را درون کد خود بنویسید که از یک سو در حین اجرا از چشمان Compiler مخفی بماند و از سوی دیگر مسئله ای را به شما یادآور شود از Comment ها استفاده می کنید. در زبان برنامه نویسی جاوا ما سه مدل Comment داریم که در بخش های آتی به تفصیل در مورد آن ها توضیح داده خواهد شد (از خط ۲ تا پایان خط ۴ نوع دیگری از آن ها قابل مشاهده است). نحوه استفاده از Comment خطی به این شکل است که در انتها الیه سمت چپ خطی از کد که نوشته اید علامت // را قرار دهید تا آن خط از کد از دید Compiler مخفی بماند (به محض اینکه علامت فوق را تایپ نمایید، رنگ کد شما به رنگ خاکستری تغییر رنگ خواهد داد و این بدان معنی است که این خط از کد به یک Comment تبدیل شده است. از این پس هر کجا که خطی یا خط هایی از کد به رنگ خاکستری کم رنگ دیدید، این بدان معنا است که آن خط یا خطوط Comment هستند). روش دیگری که برای Comment کردن خطی از کد پیشنهاد می شود این است که نشانگر موس خود را روی آن خط قرار داده و کلید کنترل به علاوه علامت / را بزنید (چنانچه این روش را روی خطی از کد اعمال کنید که از قبل Comment دارد Comment آن از بین خواهد رفت). تا اینجا شما موفق شده اید که اولین برنامه جاوای

خود را با موفقیت بنویسید. اما چنانچه از منوی اصلی گزینه Run و مجدداً گزینه Run را انتخاب کنید یا به جای این کار کلید F5 را بزنید هیچ خروجی را مشاهده نخواهید کرد چرا که در بخش main که به منزله نقطه آغازین برنامه است شما هیچ چیزی به جزء یک خط Comment که آن هم از دید Compiler مخفی می ماند ننوشته اید(در بخش های آینده در مورد تک تک واژگان خط ۵ و معنی آن ها به تفصیل توضیح داده خواهد شد). با توجه به نوع کیبورد شما ممکن است برای فعال کردن دکمه F5 نیاز باشد هم زمان دکمه Fn را هم پایین نگه دارید.

به منظور برنامه نویسی ما نیاز داریم تا اطلاعات مورد نظر خود را در جایی ذخیره سازیم. برای این کار ما اطلاعات را بسته به نوع آن ها درون یک Variable ذخیره می سازیم. در زبان برنامه نویسی جاوا ما ۸ نوع Variable داریم و نکته ای که در اینجا می بایست خیلی مد نظر قرار داده شود این است که هر Variable می تواند نوع خاصی از Data یا داده را در خود ذخیره سازد. به طور مثال، اگر Variable که مخصوص ذخیره سازی یک عدد صحیح است را برای یک تعداد واژه تعریف کنیم برنامه از ما ایراد خواهد گرفت. در طول آموزش ها با انواع Variable ها، نحوه به کار گیری آن ها، نحوه اختصاص Value یا مقدار به آن ها، به کار گیری اعمال اصلی میان آن ها مثل جمع و تفرین و ... آشنا خواهیم شد. در اولین برنامه ای که ما می نویسیم می خواهیم جمله In the Name of God به معنی به نام خدا پس از اجرای برنامه نمایش داده شود. بنابراین ما نیاز به یک Variable داریم تا یک Value از جنس کلمه، جمله یا پاراگراف را بتواند در خود جای دهد. String این کار را به راحتی می تواند برای ما انجام دهد. نکته ای که می بایست مد نظر قرار داده شود این است که هر Variable از سه بخش تشکیل می شود. بخش اول نوع خود Variable است که در این جا String می باشد(به خاطر داشته باشید که اولین حرف این واژه با حرف بزرگ نوشته شده است). بخش دوم نام

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

Variable است که این نام کاملاً اختیاری است فقط می بایست به خاطر داشته باشیم که اولین حرف نام Variable می بایست با حرف کوچک نوشته شود. در اینجا نام text را در نظر می گیریم. و در نهایت Value آن که مابین " " قرار می گیرد. به منظور اختصاص دادن یک Value به یک Variable می بایست از = استفاده کرده و در نهایت یک ; قرار داد. در واقع در زبان برنامه نویسی جاوا و برخی از دیگر زبان های برنامه نویسی ; عملکردی همچون نقطه در حین نگارش یک متن دارد. به عبارتی با قرار دادن یک ; در انتهای یک Statement به جاوا می گوییم که به پایان Statement رسیده ایم. Variable فوق را به طور خلاصه به شکل زیر نوشته می شود:

```
String text = "In the name of God";
```

حال این Variable را در بعد از { مربوط به Method با نام main قرار می دهیم.
حال کد شما می بایست به شکل زیر باشد:

```
1 class FirstProgram {  
2  
3     public static void main(String[] args) {  
4         String text = "In the name of God";  
5     }  
6  
7 }
```

این خط از کد یک Variable می باشد که هدف آن نمایش دادن جمله In the Name of God است. اما خط فوق صرفاً یک Variable است که مقداری را در خود جای داده است و برای به نمایش در آمدن نیاز به ساختاری دارد که آن را به Compiler بفرستد و نهایتاً به نمایش درآید که در بخش ذیل به توضیح نحوه اجرای این کار می پردازیم.

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

در خط بعد یک Statement با محتوای `System.out.println();` بنویسید. این خط از کد به کامپیوتر شما این دستور را می دهد که هر آنچه مابین دو پرانتز قرار دارد را به نمایش در آورد. پس در این مرحله نام Variable که در مرحله قبل ساخته بودیم را داخل پرانتز تایپ می کنیم (به خاطر داشته باشید که چنانچه نام Variable که text می باشد را با حرف بزرگ به صورت Text بنویسیم برنامه از ما خطا خواهد گرفت چرا که زبان برنامه نویسی جاوا یک زبان case-sensitive است).

یکی از ویژگی های اکثر زبان های برنامه نویسی case-sensitive بودن آن ها است به این معنی که به بزرگ و کوچک بودن حروف حساس هستند. به عبارتی اگر شما کلمه System که با حرف بزرگ نوشته شده است را با حرف کوچک به صورت system بنویسید در حین اجرای برنامه با مشکل مواجه خواهید شد چرا که System جزو Keyword ها یا اصطلاحات درونی خود جاوا است که فقط و فقط به صورت System و نه system برای جاوا قابل فهم است (در ادامه آموزش ها با انواع Keyword ها بیشتر آشنا خواهیم شد). در این مرحله شما با موفقیت توانستید اولین برنامه کامل جاوای خود را بنویسید. کدهای شما می بایست دقیقاً با کد ذیل مطابقت داشته باشد:

```
1 class FirstProgram {  
2  
3     public static void main(String[] args) {  
4         String text = "In the name of God";  
5         System.out.println(text);  
6     }  
7  
8 }
```

پیش از اجرای برنامه ما نیاز داریم تا برنامه را Compile کرده و به عبارتی به دستوراتی تبدیل کنیم که برای کامپیوترمان قابل فهم باشند (صفر و یک). حال از منوی اصلی گزینه

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

Run و مجدداً گزینه Run را انتخاب کرده، اولین برنامه شما پس از Compile شدن به صورت اتوماتیک توسط اکتیپس با موفقیت اجرا خواهد شد و در بخش Console نرم افزار اکتیپس که در قسمت پایین قرار دارد و با اجرای برنامه به صورت خود کار فعال خواهد شد مقداری را که به Variable از جنس String تحت عنوان text داده بودید را مشاهده خواهید کرد.

زمانیکه شما Value مرتبط با Variable خود تحت عنوان text را تغییر دهید و جمله دیگر را داخل " " بنویسید، Variable شما مقدار قبلی خود را کاملاً از دست داده و مقدار جدید را در خود ذخیره می سازد و از این رو پس از اجرای برنامه در بخش Console مقدار جدید اختصاص داده شده به text نمایش داده می شود.



در جلسه اول آموزش، برنامه نویسان مبتدی با تعدادی واژگان آشنا شدند که هر کدام از آن ها در زبان برنامه نویسی جاوا دارای معنی خاصی می باشند و چنانچه بخواهیم یک زبان برنامه نویسی مثل جاوا را به خوبی درک کنیم می بایست با Syntax آن زبان کاملاً آشنا باشیم. برای روشن تر شدن این مطلب، زبان برنامه نویسی جاوا را با زبانی که به آن مکالمه می کنیم مقایسه می کنیم. در واقع زبان فارسی که ما به آن تکلم می کنیم از یک دستور زبان که در برگیرنده تعدادی قاعده و قانون است و همچنین تعدادی واژه تشکیل شده است. حال این واژگان فارسی بر اساس قواعد خاصی می توانند در کنار یکدیگر

قرار گیرند تا بتوانیم از آن طریق با یکدیگر ارتباط برقرار سازیم. در حقیقت زبان برنامه نویسی جاوا هم مثل زبان فارسی از یکسری قواعد، اسم و غیره تشکیل شده است و برای آنکه بخش های مختلف یک برنامه جاوا بتوانند با یکدیگر ارتباط برقرار سازند، می بایست این اصول کاملاً مد نظر قرار داده شوند. در حقیقت در زبان برنامه نویسی جاوا اسامی در قالب ابزارهای API یا Application Programming Interface شناخته می شوند. کسانی که زبان برنامه نویسی جاوا را طرح ریزی کردند اسامی خاصی را برای بخش های مختلف زبان جاوا در نظر گرفته و در این زبان برنامه نویسی گنجانده اند. به عنوان مثال از قواعد و قوانین جاوا می توان به قرار دادن ; در انتهای هر Statement اشاره کرد که در صورتیکه فراموش کنیم آن را بنویسیم گویی فراموش کرده ایم جمله ای که در یک نامه فارسی نوشته ایم را با قرار دادن یک نقطه به پایان برسانیم (شاید اگر در نوشتن یک نامه گاهی اوقات فراموش کنیم که نقطه گذاری کنیم و هیچ مشکلی هم پیش نیاید، اما در زبان برنامه نویسی جاوا قواعد گرامری خیلی سخت گیرانه تر بوده و چنانچه فراموش کنیم یک ; در انتهای یک Statement قرار دهیم برنامه ای که نوشته ایم اجرا نخواهد شد). قواعد زبان برنامه نویسی جاوا خیلی گسترده نیستند اما در عوض API یا همان آجرهای تشکیل دهنده این زبان برنامه نویسی که در برگیرنده واژگان زبان برنامه نویسی جاوا نیز می باشند خیلی گسترده هستند و بیش از ۴۰۰۰ عنوان را شامل می شود و با انتشار هر نسخه جدید جاوا تعدادی هم به این لیست گسترده اضافه خواهد شد. اما خبر امیدوار کننده این است که شما به عنوان یک برنامه نویس جاوا نیازی به حفظ کردن کلیه این ابزارها نبوده و با توجه به نیازی که از زبان جاوا دارید همان ابزارها را به کار خواهید گرفت که در طول زمان ملکه ذهن شما خواهد شد. واژگان API جاوا به سه دسته تقسیم می شود که عبارتند از: Keyword, Identifier, API Identifier

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

در حقیقت Keyword به واژه ای در جاوا اطلاق می شود که همواره دارای معنای ثابتی است و هیچ فرقی هم نمی کند که در کجای برنامه مورد استفاده قرار گیرد. به طور مثال واژه **میز** در زبان فارسی همواره دارای معنای ثابتی است و هیچ فرقی هم نمی کند که توسط چه کسی مورد استفاده قرار گیرد. در زبان برنامه نویسی جاوا Boolean یک Keyword است و همواره داری معنای ثابتی است. از دیگر Keyword های جاوا می توان به **if, else, char** نیز اشاره کرد. در زبان جاوا Identifier به واژه ای اطلاق می شود که ما برای نامیدن چیزی در نظر می گیریم. در پروژه ساخت اولین برنامه جاوا، ما برای Variable خود نام **text** را در نظر گرفتیم. در واقع **text** یک واژه از نوع Identifier می باشد و این در حالی است که ما می توانستیم به جای واژه **text** از واژگان دیگری همچون **sample, tester, myString, aliString** نیز استفاده کنیم. چنانچه بخواهیم زبان برنامه نویسی جاوا را با زبان فارسی مقایسه کنیم، می توان گفت که واژگانی از قبیل اشکان، کیانا، احسان و غیره از نوع Identifier می باشند. حال نوبت به توضیح پیرامون API Identifier می رسد که امیدواریم با این توضیح مفهوم آن کاملاً روشن گردد. در زبان فارسی یکسری واژگان هستند که از نوع Identifier می باشند و به عبارتی یکسری نام هستند که به چیزی اطلاق می شوند اما این گروه از واژگان در طول زمان نهادینه شده و پس از مدتی صرفاً به یک چیز خاص اطلاق می شوند مثل مولوی که ابتدأً یک واژه از نوع Identifier بوده اما پس از گذشت زمان به یک Identifier خاص تبدیل شده که فقط و فقط به شاعر بزرگ ایرانی اطلاق می شود. در زبان برنامه نویسی جاوا تعدادی واژه وجود دارند که اصطلاحاً به آن ها API Identifier گفته می شود. درست است که این گروه از واژگان یک واژه معمولی هستند ولی از آنجا که پایه ریزان زبان برنامه نویسی جاوا به کرات از آن ها استفاده کرده اند تبدیل به واژگانی شده اند که فقط و فقط به یک چیز خاص اشاره

دارند و عملکردی همانند واژه مولوی دارند و این گروهی از این واژگان عبارتند از
String, main, System, out, println

یکی دیگر از مواردی که در جاوا خیلی مورد استفاده قرار می گیرد مفهوم Method است. در واقع Method به لیستی از کارهایی گفته می شود که قرار است به ترتیب خاص انجام شوند. در زبان برنامه نویسی جاوا هر Method دارای یک نام است که با آوردن نام متد در جایی از برنامه از جاوا می خواهیم که لیست کارهایی که درون متد برنامه ریزی شده اند را انجام دهد.

```
1 class FirstProgram {  
2  
3     public static void main(String[] args) {  
4         String text = "In the name of God";  
5         System.out.println(text);  
6     }  
7  
8 }
```

کدهای فوق مربوط به مثال اولین پروژه جاوا است که طراحی کردیم. به منظور روشن تر شدن مطالب فوق الذکر، هر یک از اصطلاحات را از روی کد مشاهده خواهیم کرد. در این کد در خط اول یک Class با نام FirstProgram داریم و علامت { که پس از نام Class آمده و علامت } که در خط هشتم آمده است تشکیل دهنده یک block است که کل برنامه ما را در بر می گیرد. در خط سوم یک Method با نام main داریم که حاوی لیستی از دستورات می باشد. واژگانی مثل public, static, void در بخش های آتی توضیح خواهیم داد. در واقع هر زمانیکه ما یک Method با نام main ایجاد کنیم، به Compiler این دستور را می دهیم که این بخش از کد بخشی است که برنامه را شروع می کند و در حین اجرا کامپیوتر به صورت خود کار برنامه نوشته شده توسط ما را از این بخش از برنامه به بعد اجرا می نماید. علامت { که پس از Method قرار گرفته و علامت } در خط ششم به منزله block دوم برنامه ماست. در

این block ما یکسری دستورات مبنی بر نوشتن جمله In the Name of God روی مانیتور نوشته ایم که به ترتیب آن ها را مورد بررسی قرار خواهیم داد. در این Method ما یک Variable از جنس String تحت عنوان text داریم. در این Statement واژه String یک API Identifier است و واژه text یک Identifier می باشد. در خط پنجم ما یک Statement دیگر مشاهده می کنیم که حاوی واژگان System, out, println می باشد که همگی جزو API Identifier ها می باشند.