

**چگونه از C++ به جاوا
مهاجرت کنیم!!!**

نویسنده :

سعید رضا ولی زاده

srvalizadeh@yahoo.com

مقدمه

ویژگی های منحصر به فرد جاوا مانند شیء گرایی ، API های فراوان ، کتابخانه نرم افزاری بزرگ و همچنین مستقل بودن اجرای برنامه های جاوا از سیستم عامل ، باعث شده که روز به روز بر تعداد برنامه نویسان این زبان افزوده شود.

امروزه بسیاری از برنامه نویسان C++ تمایل دارند که از این زبان به سمت Java مهاجرت کنند. شاید بتوان گفت این مهاجرت بسیار ساده هست زیرا که دستورات و نحوه برنامه نویسی در زبان جاوا بسیار شبیه به C++ است و فقط کافی است تفاوت هایی که بین این دو زبان وجود دارد را یاد بگیرید. البته قابل ذکر است این کار به سادگی خواندن این مقاله نیست ، اما این مقاله نقطه شروعی برای جاوا نویس شدن است و تا تبدیل به جاوا کار شدن حرفه ای راه زیادی در پیش است.

برای شروع ، ابتدا یک برنامه ساده را در جاوا نوشته ، کامپایل و اجرا می کنیم و در ادامه به صورت علمی تر به نوشتن برنامه در جاوا می پردازیم.

در این مقاله سعی بر این بوده که مراحل این مهاجرت به صورت تصویری و گام به گام و با زبانی ساده توضیح داده شود. امیدوارم بتوانید از این مقاله استفاده لازم را ببرید. در صورت داشتن هرگونه مشکل با پست الکترونیکی srvalizadeh@yahoo.com مکاتبه کنید.

گام ۱- قبل از شروع (قبل از انجام هر کاری نکات زیر را یکبار مطالعه کنید تا زمینه ای برای شروع داشته باشید).

- ✓ نکته ۱- همان طور که می دانید جاوا طوری طراحی شده که برای برنامه نویسی نیاز به هیچ محیط خاصی ندارد. البته استفاده از یک IDE مناسب مانند Net Beans ، JBuilder ، eclipse و... کار را برای شما بسیار آسان می کند اما شروع کار با این IDE ها ممکن است شما را تا حد زیادی سردرگم کند. توصیه من برای کسانی که تازه قصد مهاجرت را دارند این است که فعلا برنامه های خود را در یک ویرایشگر متن ساده مانند Notepad ویندوز بنویسند و فایل برنامه شان را با پسوند java ذخیره کنید. برای ویرایش کردن این فایل روی فایل راست کلیک کنید و Edit را انتخاب کنید.
- ✓ نکته ۲- جاوا یک زبان حساس به متن است یعنی نسبت به حروف کوچک و بزرگ حساس است. پس حتما در نوشتن نام متد ها ، نام کلاس ها ، نام متغیر ها و ... به این نکته توجه کنید.
- ✓ نکته ۳- همان طور که گفته شد جاوا یک زبان کاملا شیء گرا است پس تمام روند برنامه نویسی شما حتما و حتما باید داخل یک کلاس صورت بگیرد.!!!!!!
- ✓ نکته ۴- تمام روند برنامه در جاوا داخل یک کلاس اصلی قرار می گیرد که نام این کلاس با نام فایل محتوی کد برنامه دقیقا باید یکسان باشد. به عنوان مثال اگر نام کلاس اصلی HelloWorld باشد نام فایل شما باید به صورت HelloWorld.java باشد (به بزرگ تایپ شدن حرف H و W توجه کنید).

گام ۲ - قالب اصلی یک برنامه در جاوا

در جاوا قالب اصلی هر برنامه به صورت زیر است

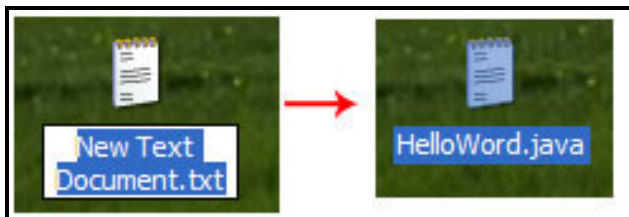
```
public class [نام کلاس] {  
    public static void main(String[] args) {  
        }  
}
```

همان طور که قبلا توضیح داده شد ، به جای عبارت [نام کلاس] باید همان نام کلاسی را نوشت که کلاس های دیگر را در بر می گیرد که این نام با نام فایل در بردارنده کد برنامه باید یکسان باشد.

- ✓ به عنوان یک نکته قابل ذکر است که در جاوا به توابع متد می گوئیم.
- ✓ متد main را دقیقا مانند قالب بالا تعریف کنید.

گام ۳- شروع برنامه نویسی :

تمام کتاب های آموزش جاوا برای اولین مثال برنامه نویسی ، مثال " کلمه سلام " را معرفی می کنند. ما نیز با همین مثال کار را شروع می کنیم. همان طور که در تصویر زیر مشخص است یک فایل Text جدید با عنوان HelloWorld بسازید و پسوند آن را به java. تغییر دهید تا نام فایل به صورت HelloWorld.java در آید.



(در صورتی که در ویندوز شما پسوند ها فعال نیستند وارد My computer شده از منو tools گزینه Folder Option را انتخاب کنید. در پنجره باز شده وارد تب View شده و تیک مربوط به گزینه Hide Extensions for know file type را بردارید.)

حال روی فایل ساخته شده راست کلیک کرده و گزینه Edit را انتخاب کنید تا وارد محیط Notepad شوید.

گام ۳- شروع کد نویسی (اولین مثال)

همان طور که در تصویر پایین صفحه می بینید در این گام باید کد زیر را در فایل text باز شده تایپ کنید.

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

تغییرات را در فایل ذخیره کنید.

توجه داشته باشید که در این مثال بلوک کلاس HelloWorld ، بلوک تمام کلاس های دیگر را در بر گرفته است.

(منظور از بلوک علامت های { و } هستند.)

✓ این برنامه کلمه Hello Word را چاپ می کند. (توضیحات مربوط به دستورات جاوا را در قسمت بعد

توضیح خواهم داد.)

A screenshot of a Notepad window titled 'HelloWord.java - Notepad'. The window contains the following code:

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

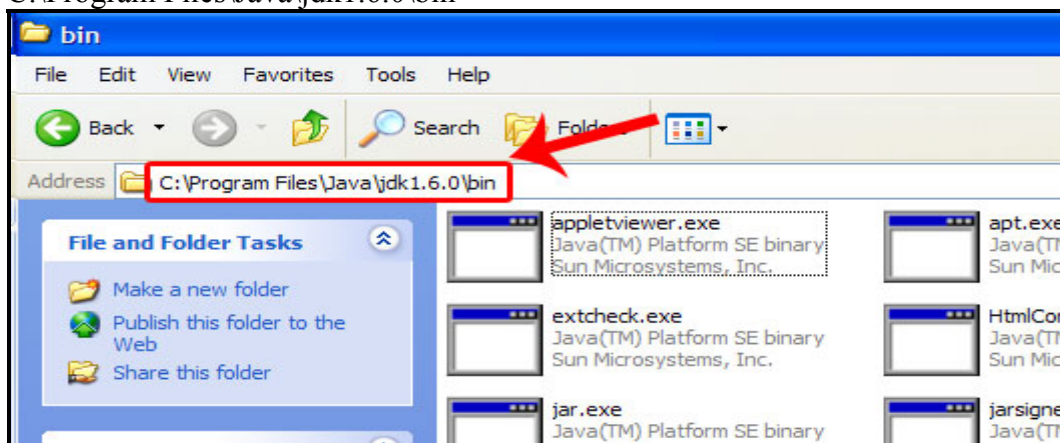
گام ۳ – آماده سازی برنامه برای کامپایل (فایل HelloWorld.java)

ما برای کامپایل کردن برنامه نیاز به Java Development Kit یا به اختصار JDK داریم. (متأسفانه با اینکه JDK رایگان است اما به علت تحریم ایران، شرکت Sun اجازه دانلود jdk را از سایت رسمی خود نمی دهد و باید jdk را از سایت های غیر رسمی که jdk را Share کرده اند دانلود کنید.) آخرین ورژن JDK در زمان تالیف این مقاله 1,6 یا به اختصار ۶ است.

(مراحل نصب JDK آنقدر ساده است که از توضیح آن صرف نظر می کنم!)

پس از نصب jdk شما باید وارد پوشه bin از jdk شوید. به عنوان مثال اگر شما jdk را در دیسک C نصب کرده باشید باید به آدرس زیر بروید:

C:\Program Files\Java\jdk1.6.0\bin



(توجه کنید که ورژن jdk شما می تواند متفاوت باشد و نام پوشه ها و محل آن ها تا حدی تغییر کند!)

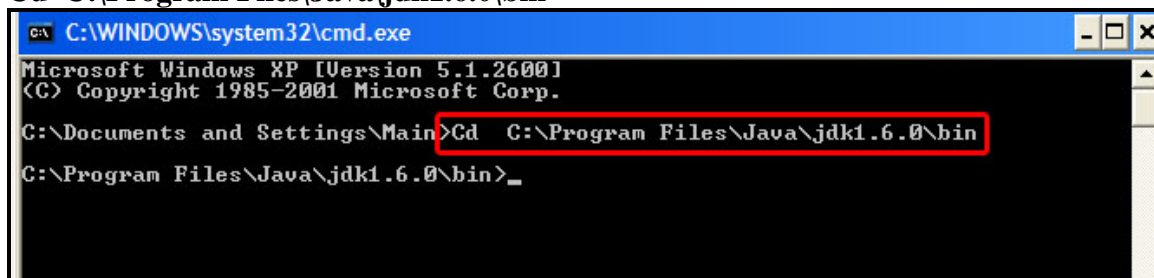
برای کامپایل هر فایل دو مرحله زیر را انجام دهید:

الف) فایلی را که قبلاً ساخته اید را در پوشه bin از jdk کپی کنید. در اینجا فایل HelloWorld.java را کپی می کنیم

ب) وارد Command Prompt شوید (Start >> Run >> CMD)

و با دستور cd به همان دایرکتوری bin بروید یعنی:

Cd C:\Program Files\Java\jdk1.6.0\bin



(برای راحتی کار در Command Prompt، موقع رفتن به یک آدرس، کافی است که یک یا چند حرف اول نام هر پوشه را تایپ کنید و با زدن کلیک TAB ویندوز نام آن پوشه را تکمیل می کند)

گام ۴- کامپایل کردن

دستور کامپایل برنامه Javac می باشد.

حال می توانید برنامه خود را با دستور Javac با الگوی زیر کامپایل کنید

Javac [filename.java]

که به جای [filename.java] نام فایل با پسوند .java را قرار می دهیم

به عنوان مثال برای کامپایل فایل HelloWorld.java به صورت زیر عمل می کنیم:

Javac HelloWorld.java

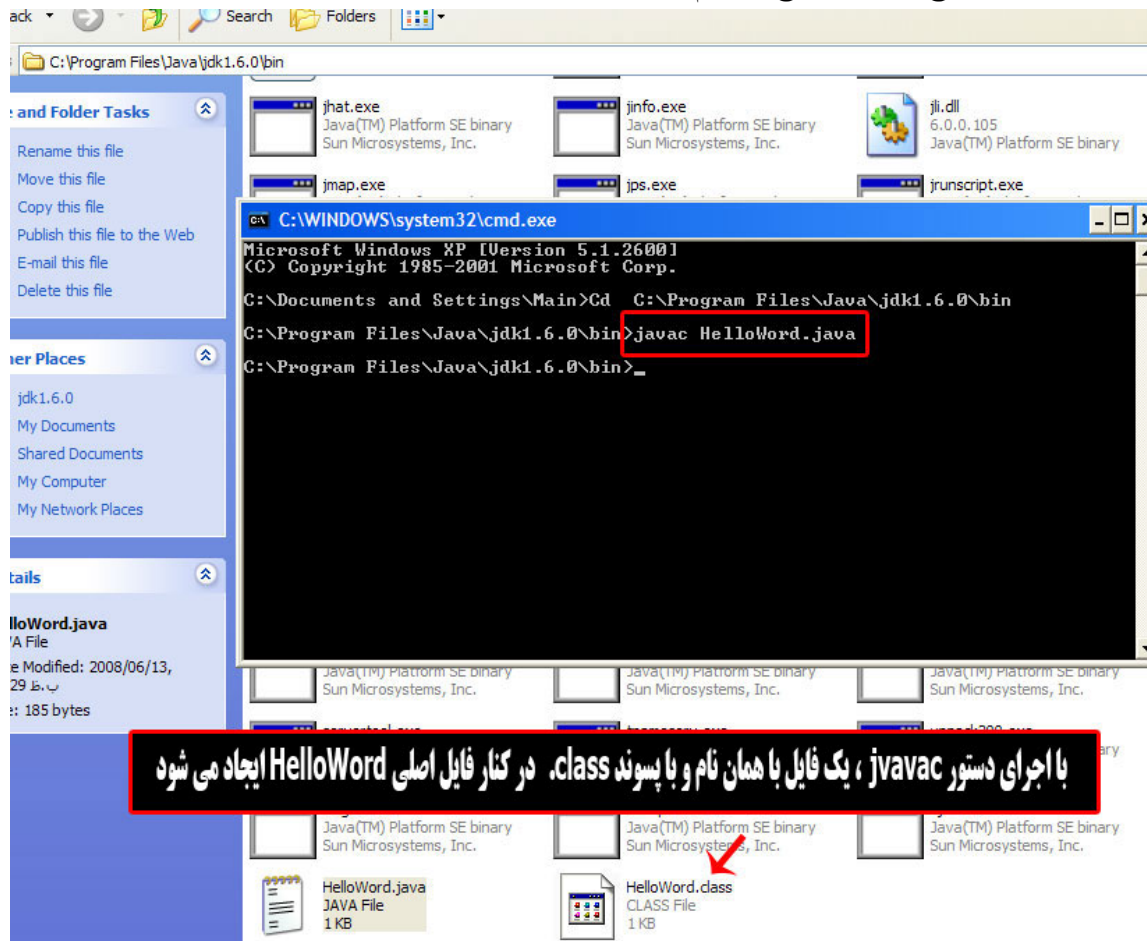
(باز هم توجه کنید که کوچک و بزرگ بودن حروف باید رعایت شود!)

همان طور که در شکل زیر می بینید در صورتی که برنامه شما هیچ Error نداشته باشد jdk یک فایل با همان نام ولی

با پسوند .class در دایرکتوری bin ایجاد می کند

✓ در صورتی که کد شما خطایی داشته باشد jdk محل و نوع خطا را به طور دقیق به شما گزارش می دهد. (در

مورد خطاهای رایج در ادامه توضیح می دهیم)



گام ۵- اجرا کردن برنامه

دستور اجرای برنامه Java می باشد.

برای اجرا کردن برنامه در همان command prompt دستور java را با الگوی زیر اجرا می کنیم

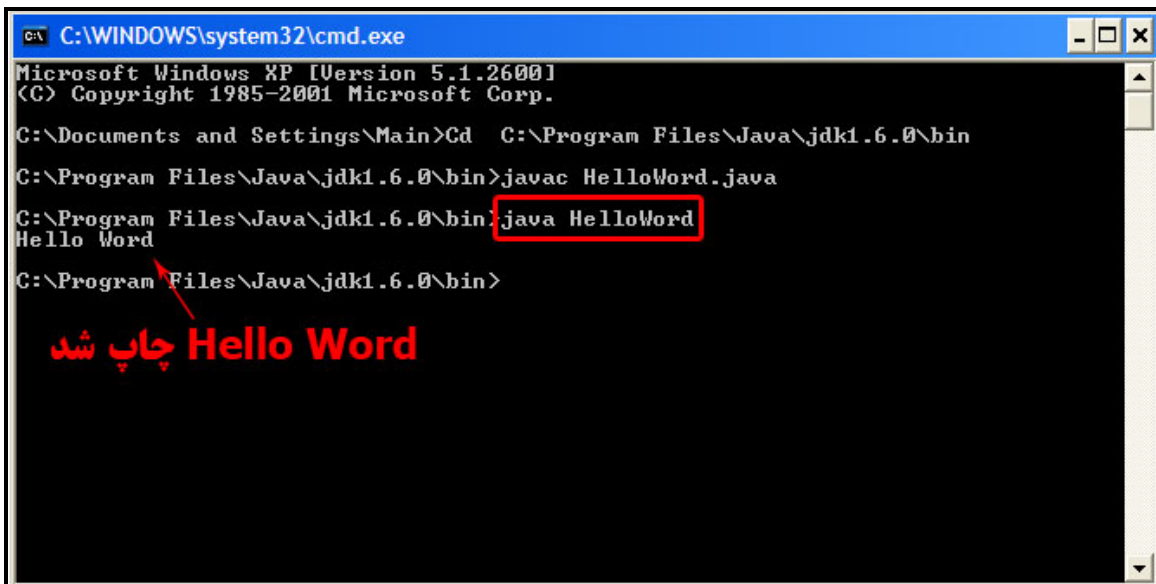
Java [filename]

✓ توجه کنید که در دستور Java [filename] بر خلاف دستور Javac [filename.java] دیگر پسوند فایل را نباید ذکر کنید.

که به جای [filename] نام فایل را بدون هیچ پسوندی را قرار می دهیم
به عنوان مثال برای اجرا کردن فایل HelloWorld.java به صورت زیر عمل می کنیم:

Java HelloWorld

با این عمل برنامه اجرا شده و کلمه Hello word در command prompt چاپ خواهد شد.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Main>Cd C:\Program Files\Java\jdk1.6.0\bin
C:\Program Files\Java\jdk1.6.0\bin>javac HelloWorld.java
C:\Program Files\Java\jdk1.6.0\bin>java HelloWorld
Hello Word
C:\Program Files\Java\jdk1.6.0\bin>
```

Hello Word چاپ شد

در صورتی که نتوانسته اید و با Error مواجه شده اید مطمئن باشید که یکی از مراحل را به درستی انجام نداده اید. برای درک معانی خطاها در جاوا می توانید به مقاله "[مشکلات رایج در جاوا](http://java4every1.wordpress.com)" از وبلاگ <http://java4every1.wordpress.com> مراجعه کنید. این وبلاگ آموزش های بسیار خوبی به صورت فارسی در جاوا دارد.

خسته نباشید...! تا اینجا باید اولین برنامه تان در جاوا را نوشته ، کامپایل و اجرا کرده باشید. حال به تفاوت های دستورات بین ++C و جاوا می پردازیم.

تفاوت های کد نویسی :

تفاوت ۱- چاپ : در C++ برای چاپ دستوری به نام `cout` داشتیم اما در جاوا به جای `cout` از دستور زیر استفاده می کنیم:

```
System.out.println("این نوشته چاپ می شود");
```

مثلا برای چاپ یک رشته مانند Hello Java به صورت زیر عمل می کنیم

```
System.out.println("Hello Java");
```

(توجه داشته باشید که دستورات دیگری نیز برای چاپ وجود دارد که فعلا به دستور بالا کفایت می کنیم.

تفاوت ۲ - تعریف فایل Header

همان طور که می دانید در هر زبانی برای اینکه بتوان از توابع از پیش نوشته شده استفاده کرد باید فایل کتابخانه ای مربوط به آن تابع را به برنامه اضافه کرد. مثلا در C++ برای اینکه بتوان از دستور `cout` استفاده کرد باید فایل `IOStream.h` را به ابتدای برنامه ضمیمه کرد. یا برای استفاده از تابع `getch` فایل `stdio.h` را اضافه می کنیم. اینکار در C++ با دستور `include` به صورت زیر انجام می شد

```
#include <iostream.h>
#include <conio.h>
```

اما در جاوا با دستور `import` این کار را انجام می دهیم. الگوی این دستور به صورت زیر است:

```
import [نام کتابخانه] ;
```

مثلا برای اینکه بتوانیم از توابع مربوط به آرایه ها استفاده کنیم باید دستور زیر را بنویسیم

```
import java.util.Arrays;
```

تفاوت ۳- تعریف آرایه

در C++ آرایه را هم می توانستیم به صورت دینامیک و هم به صورت استاتیک تعریف کنیم اما جاوا آرایه را فقط به صورت دینامیک تعریف می کند. شکل کلی تعریف آرایه در جاوا به صورت زیر است:

```
[طول آرایه][نوع آرایه] = new نام آرایه [نوع آرایه];
```

مثلا اگر بخواهیم آرایه ای با طول ۱۰۰ و با عنوان `a` از نوع `int` تعریف کنیم به صورت زیر عمل می کنیم:

```
int[] a = new int[100];
```

شاید تفاوت های زیادی دیگری نیز وجود داشته باشد که شما در مراحل کد نویسی های پیشرفته به آن پی ببرید ، اما این ۳ تفاوت ، تفاوت هایی است که شما در هر کد برنامه جاوا مشاهده خواهید کرد.

در موقع بررسی تفاوت ها به مقاله ای به زبان انگلیسی برخورد کردم تحت عنوان A Crash Course from C++ to Java . جهت مطالعه علمی تر و بررسی دقیق تر مهاجرت از C++ به جاوا می توانید این مقاله را از آدرس زیر دانلود کنید

http://www3.interscience.wiley.com:8100/legacy/college/horstmann/0471164372/chapter/c_to_java.pdf

با آرزوی موفقیت و مفید بودن این مقاله برای شما.

پایان