



# آموزش برنامه‌نویسی جاوا

قسمت دوم

مدرس: آقای چگانی

[ILikePHP.ir](http://ILikePHP.ir)



ILIKEPHP.IR



سلام، چگانی هستم و با قسمت دوم آموزش جاوا در خدمتون هستم.

قسمت اول آموزش رو میتونید از آدرس زیر بخونید:

<http://ilikephp.ir/index.php?r=news&id=6>

در صورتی که مشکلی یا سوالی از جلسه قبل دارین بنده در خدمتم.

خوب پس بریم سراغ شروع جلسه دوم

قبل از اینکه وارد بحث اصلی آموزشمون بشیم یه مثال خیلی خیلی ساده از جاوا اجرا کنیم. قبل از اجرا شما لازم هست JDK را نصب کنید. JDK ابزار توسعه و کامپایل کدهای جاوا هست. همچنین برای کدنویسی پیشنهاد میکنم IntelliJ IDEA نصب کنید که لینکهای اون به این صورت هستند:

[/https://www.jetbrains.com/idea/download](https://www.jetbrains.com/idea/download)

[/http://p30download.com/fa/entry/33429](http://p30download.com/fa/entry/33429)

توصیه میکنم نسخه ۱.۸ jdk نصب شود.



که مراحل نصب و راه اندازی رو میسپارم به دوستان و اگه مشکلی بود تا جلسه بعدی بپرسین حتما

خوب بهتر هست قبل از آموزش اول دو مثال خیلی خیلی ساده از جاوا ببینیم.

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

خوب بریم سراغ کد نویسی:

این کد در واقع متن Hello رو داخل کنسول یا همون output برای شما چاپ میکنه.

تابع main که همیشه به همین صورت ظاهر میشه در واقع اولین نقطه اجرا کدتون هست. یعنی همیشه کدتون با تابع main

شروع میشه و امضای این تابع دقیقا همیشه به این شکل هست:

```
public static void main(String args[])
```

که بعدا در مورد بخش های مختلفش صحبت می کنیم.



نکته بعدی اینکه دقت کنید تابع main داخل یک کلاس قرار دارد. در واقع در جاوا هر کلاس در یک فایل قرار میگیرد که همیشه اسم فایل و اسم کلاس باید یکی باشد. در اینجا ما این کد رو داخل فایل Main.java نوشتیم.

نکته بعدی اینکه جاوا case sensitive هست یعنی به کوچک و بزرگ بودن حروف حساس هست.

نکته بعدی اینکه توی زبان های شی گرا از کلمه تابع استفاده نمی کنند و بجاش از کلمه متد استفاده میشه. یعنی در واقع تابع

main از این به بعد همیشه متد !!main

خوب الان یه مثال دیگه با هم ببینیم تا بیشتر در جریان قرار بگیریم:

```
Main.java x Main2.java x
1  import java.util.Scanner;
2
3  public class Main2 {
4      public static void main(String args[]) {
5          Scanner input = new Scanner(System.in);
6          int number1;
7          int number2;
8          int sum;
9          System.out.print("Enter first integer: ");
10         number1 = input.nextInt();
11         System.out.print("Enter second integer: ");
12         number2 = input.nextInt();
13         sum = number1 + number2;
14         System.out.print("Sum is =" + sum);
15     }
16 }
```



مدرس : آقای چگانی

آموزش JAVA قسمت اول

[Mohammad Changani, [10.09.15 12:37

توی این کد ما دو عدد از کاربر (از قسمت output) دریافت کردیم باهم جمع کردیم و در خروجی چاپ کردیم. ولی چطوری؟

[Mohammad Changani, [10.09.15 12:38

فایل جدیدی به نام Main2.java ساختیم و داخل اون کلاس Main2 رو میسازیم. برای اینکه هر بار نخواین امضای main رو بنویسید کافیه بنویسید psvm و دکمه tab کیبرد رو بزنید! خود IDE تکمیل میکنه!

[Mohammad Changani, [10.09.15 12:38

خوب توی خط ۵ ما داریم نمونه‌ای به نام input از کلاس Scanner میسازیم تا بتونیم ازش استفاده کنیم. (عجله نکنید بعدا توضیح میدیم این نمونه ساختن به چه نحوه‌ی هست)

[Mohammad Changani, [10.09.15 12:38

توی خط ۹ متنی را چاپ میکنیم و در خط ۱۰ یک عدد از کاربر دریافت می‌کنیم و داخل متغیر number1 میریزیم. و در آخر هم حاصل را چاپ میکنیم.

[Mohammad Changani, [10.09.15 12:39



مدرس : آقای چگانی

آموزش JAVA قسمت اول

خوب دوستانی که تازه شروع کردن میتونند

برای جلسه بعدی برنامه‌آی بنویسید که ابتدا یک عدد بگیرد بعد یکی از علامت‌های + - \* و بعد عدد دوم، و حاصل این دو عدد با توجه به علامت گرفته شد هرا چاپ کنه.

هدف ای گفتن این دوتا مثال فقط این بود یکم آشناییت پیدا کنید

خوب بریم سراغ بحث اصلی این جلسه که در مورد ۲ اصل اول بحث گذشته یعنی abstraction صحبت کنیم.

خوب قبل از اینکه شروع کنیم سوالی ازتون میپرسم که بهش فک کنید. توی مثال بالا Scanner یک موجودیت بود یا یک کلاس؟ (با توجه به جلسه قبلی میتونید پاسخ بدین)

در موردش فک کنیم اخر جلسه صحبت کنیم

خوب، مهمترین نکته‌ای که در طراحی شی‌گرا باید بهش دقت داشته باشیم فهمیدن موجودیت‌های سیستم (پروژه‌ای که به ما داده شده) هست. در واقع موجودیت‌ها همان عامل‌های انتزاعی هستند که جلسه اول در مورد اون‌ها صحبت کردیم. پس قدم اول ما پیدا کردن موجودیت‌های سیستم هست.

مثال: موجودیت‌های سیستم ثبت‌نام دانشگاه

۱- دانشجو

۲- درس

۳- استاد



سوالی که پیش میاد اینکه آیا واقعا دانشگاه موجودیت هست؟

با اینکه ما فقط یک نمونه از اون رو توی سیستم داریم. جواب این سوال بله هست. موجودیت بودن یا نبودن رو تعداد نمونه های اون نشون نمیده.

پیدا کردن موجودیتها و مدل کردن سیستم کاری هست که به مرور زمان و تجربه خیلی ساده تر میشه ولی نکاتی داره که باید توی طراحی در نظر گرفت و خیلی مهم هستند که با توجه به اینکه این آموزش، آموزش جاوا هست ما روی این موضوع وقت نمیگذاریم.

بعد از فهمیدن موجودیتهای سیستم لازم هست خصیصه ها یا ویژگی های موجودیت و توانایی آنها را پیدا کنیم. از این به بعد به خصیصه ها field می گیم و تواناییها همون متدهای موجودیت هستند.

خوب پس ما باید برای تک تک این موجودیتها field ها رو مشخص کنیم و بعد بریم سراغ متدها

خوب پس کلاسی به نام Student میسازیم به این صورت:

```
1  /**
2   * Created by Mohammad Changani on 9/10/15.
3   */
4  public class Student {
5      int id;
6      String name;
7      String family;
8      String username;
9      String password;
10 }
11 |
```



تا اینجا ما این field ها را برای دانشجو در نظر گرفتیم. شاید بعدا بهش اضافه کنیم.

خوب یک توانایی یا متد به نام `printName` اضافه می کنیم که اسم و فامیل را برای ما چاپ می کند به این صورت:

```
Main.java x Main2.java x Student.java x
1  /**
2   * Created by Mohammad Changani on 9/10/15.
3   */
4   public class Student {
5       int id;
6       String name;
7       String family;
8       String username;
9       String password;
10
11      public void printName(){
12          System.out.println(name + " " + family);
13      }
14  }
15  |
```

خوب الان بریم ببینیم چطوری میتونیم از این موجودیت دانشجو استفاده کرد. من نمونه ای به نام `ali` از موجودیت دانشجو به این صورت می نویسم.

```
Student ali = new Student();
```

این عبارت همون طور که میبینید از ۴ جز اصلی تشکیل شده است. از سمت چپ به راست:





مدرس : آقای چگانی

آموزش JAVA قسمت اول

Student در واقع نوع موجودیت رو مشخص می کند و **ali** اسم نمونه هست که شما میتونید هر اسمی خواستین بگذارین. در واقع تا اینجا شبیه ساختن یه متغیر از جنس **int** هست:

```
int a;
```

که در واقع متغیری به نام **a** داریم از جنس **int** میسازیم.

کلمه **Student** در واقع نوع موجودیت رو مشخص می کند و **ali** اسم نمونه هست که شما میتونید هر اسمی خواستین بگذارین. در واقع تا اینجا شبیه ساختن یه متغیر از جنس **int** هست:

```
int a;
```

که در واقع متغیری به نام **a** داریم از جنس **int** میسازیم.

من یه نمونه دیگه به نام **std1** هم از کلاس **Student** میسازم:

```
Student std1 = new Student;()
```

خوب ما میتونیم به این صورت به اجزای یک نمونه دسترسی داشته باشیم:

```

1  /**
2   * Created by Mohammad Changani on 9/10/15.
3   */
4  public class Main3 {
5      public static void main(String[] args) {
6          Student ali = new Student();
7          Student std1 = new Student();
8
9          ali.id = 1;
10         ali.name = "ali";
11
12         std1.id = 2;
13         std1.family = "Rezai";
14
15
16         System.out.println(ali.id);
17         System.out.println(ali.name);
18         System.out.println(ali.family);
19
20         ali.printName();
21
22
23         System.out.println(std1.id);
24         System.out.println(std1.family);
25
26
27         std1.printName();
28     }
29 }
30

```

ما توی خط ۹ ویژگی id نمونه ali را مقدار ۱ قرار دادیم و در خط بعدی برای ویژگی name نمونه ali مقداری رو مشخص کردیم. خط ۱۲ و ۱۳ برای نمونه std1 و توی خط های انتهای مقدار آن ها را چاپ کردیم.



توی خط ۲۰ هم متد `printName` رو فراخوانی کردیم که درواقع اسم و فامیل کاربر را برای ما چاپ میکنه

فقط دقت کنید توی خط ۱۸ ما داریم `field` رو مقدارش رو چاپ میکنیم که مقدار دهی نکردیم بنابراین جاوا مقدار پیش فرض یعنی `null` رو برای ما چاپ میکند.

نکته که همه دوستان می دونند ولی لازم هست بگم اینکه هر نمونه ویژگی های خودش رو داره و جایگاه ذخیره سازی اون ها در حافظه متفاوت هست. یعنی `id` نمونه اول با `id` نمونه دوم و همه نمون های دیگر متفاوت هست.

تا اینجا برای این جلسه کافی هست. جلسه بعدی یا میگیریم این موجودیت رو چطور کپسوله کنیم و مطالب مرتبط با اون رو میگیریم.

میدونم خیلی خیلی بحثها ابتدایی هستند ولی کم کم سرعت یادگیری رو بالا میبریم که همه بتونند استفاده کنند پس کمی صبور باشین. به جاهای خوب هم میرسیم!

فقط همون طور قول داده بودم اخر هر جلسه چندتا از نکات نگارشی توی زبان جاوا رو میگم. دقت داشته باشید که رعایت نکردن این موارد تاثیری در کامپایل شدن یا نشدن کدتون ندارد ولی برای نوشتن استاندارد لازم هست رعایت بشه.

1) نام کلاسها با حرف بزرگ شروع می شوند. اگر نام کلاس شامل دو کلمه یا بیشتر باشد حرف اول کلمات را بزرگ می نویسم.

`Test , StudentTest, MyStudent`

2) نام متغیرها و نام متدها با حرف کوچک شروع می شوند و اگر چند کلمه ای بود، حرف اول کلمات بعدی بزرگ نوشته می شوند

`test, testStudent, printName(), name, id`



مدرس : آقای چگانی

آموزش JAVA قسمت اول

3) از - یا \_ یا \$ و کلا از سایر علامت ها برای انتخاب اسم کلاس، متد، متغییر استفاده نکنید.

[www.ILikePHP.ir](http://www.ILikePHP.ir)