

## به نام خدا

### آموزش چهل و هشتم

اهداف آموزشی این قسمت عبارتند از:

۱. ساخت شیء از روی یک کلاس
۲. آشنایی عملی با کلید واژه static
۳. نحوه به کار گیری کلید واژه static با متدها
- ۴.

در ادامه آموزش قسمت چهل و هفتم، در این آموزش قصد داریم تا کلاس دیگری ایجاد کرده آنرا به عنوان کلاس شروع برنامه خود در نظر بگیریم و در نهایت اقدام به ساخت شیء هایی از روی کلاس Athlete در کلاس جدید خود خواهیم کرد.

برای ادامه آموزش، پروژه ای که در قسمت پیش تحت عنوان 47<sup>th</sup> Session ایجاد کردیم را مد نظر قرار می دهیم. حال کلاس دیگری تحت عنوان ActionClass به معنی "کلاس عملیاتی" ایجاد کنیم و خاطرمأن باشد که در حین ساخت آن حتماً گزینه public static void main را تیک دار می کنیم چرا که همانطور که قبلاً آموزش داده شد هر کجا که متد main قرار گیرد به منزله نقطه شروع برنامه ما خواهد بود و ما هم نیاز داریم تا برنامه خود را از داخل این کلاس آغاز کنیم:

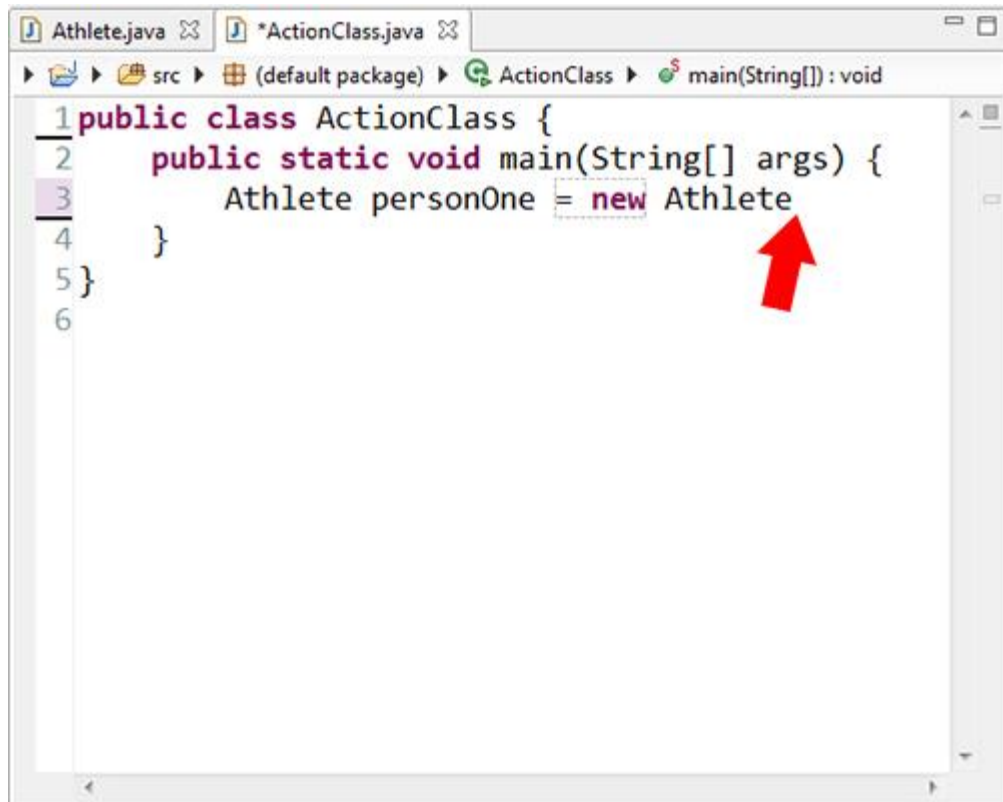
```
public class ActionClass {  
    /**  
     * @param args  
     */  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
    }  
}
```

لازم به ذکر است که برای سهولت کار می توان کامنت های فوق را حذف کرد. اکنون در بخش متد main می خواهیم یک شیء از روی کلاس Athlete ایجاد کنیم. برای این منظور کد فوق را به شکل زیر تکمیل می کنیم:

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی



```
1 public class ActionClass {
2     public static void main(String[] args) {
3         Athlete personOne = new Athlete
4     }
5 }
6
```

همانطور که در تصویر فوق مشاهده می شود برای ساخت یک شیء جدید از روی کلاس Athlete اول واژه Athlete را نوشته سپس نامی برای آن در نظر می گیریم که در اینجا نام personOne به معنی "نفر اول" را در نظر گرفته ایم. سپس یک علامت مساوی قرار داده و کلید واژه new را می نویسیم و پس از قرار دادن یک فاصله مجدد نامی کلاسی که می خواهیم از روی آن یک شیء ایجاد کنیم را می نویسیم. همانطور که در تصویر فوق مشاهده می شود انتهای واژه Athlete با یک فلش قرمز رنگ مشخص شده است. در واقع با قرار دادن نشانگر موس خود در آن نقطه و فشردن هم زمان کلید های Ctrl و Space با تصویر زیر مواجه خواهیم شد:



همانطور که در تصویر فوق مشاهده می شود از آنجا که برای Constructor خود ما سه پارامتر در نظر گرفته ایم، اکلیپس این نکته را متوجه شده و به صورت خود کار شیئی ما را به همراه نام پارامترها تکمیل می کند. اکنون می بایست نام پارامترهای فوق را بسته به اینکه چه جنسی دارند با مقادیر متناظر با آنها تکمیل کنیم. برای این منظور کد خود را به شکل زیر تکمیل می کنیم:

```

public class ActionClass {
    public static void main(String[] args) {
        Athlete personOne = new Athlete("Behzad", "Moradi", 1362);
    }
}

```

در حقیقت از آنجا که جنس دو پارامتر اول String است از اینرو مقدار آنها را می بایست داخل دو علامت " " قرار داد و از آنجا که جنس پارامتر سوم int است می بایست یک عدد صحیح برای آن در نظر گرفت که در مثال فوق عدد ۱۳۶۲ برای آن در نظر گرفته شده است. اکنون نیاز داریم تا به متد قرار گرفته در کلاس Athlete دست پیدا کنیم. برای این منظور کد فوق را به صورت زیر تکمیل می کنیم:

```

public class ActionClass {
    public static void main(String[] args) {
        Athlete personOne = new Athlete("Behzad", "Moradi", 1362);
        personOne.showAthleteInfo();
    }
}

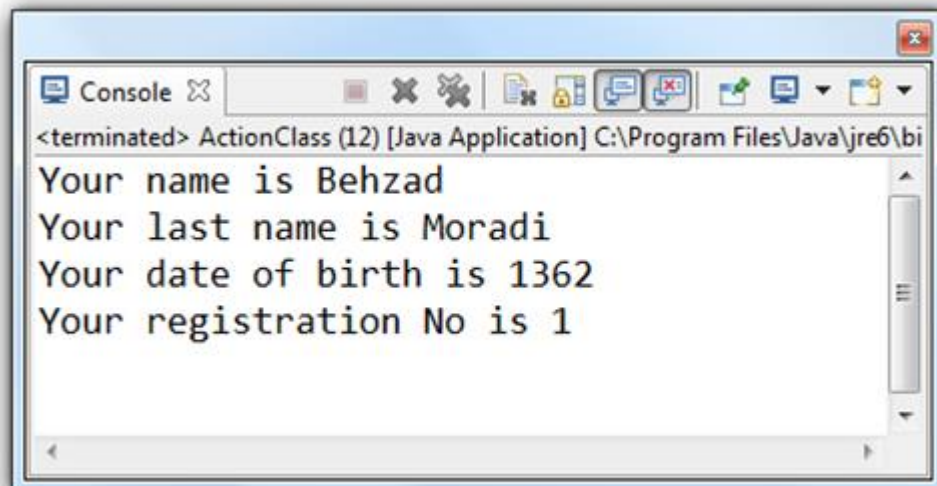
```

همانطور که قبلاً گفته شد این متد وظیفه دارد تا اطلاعات هر ورزشکار را به نمایش در آورد. حال می توانیم برنامه خود را اجرا کنیم:

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

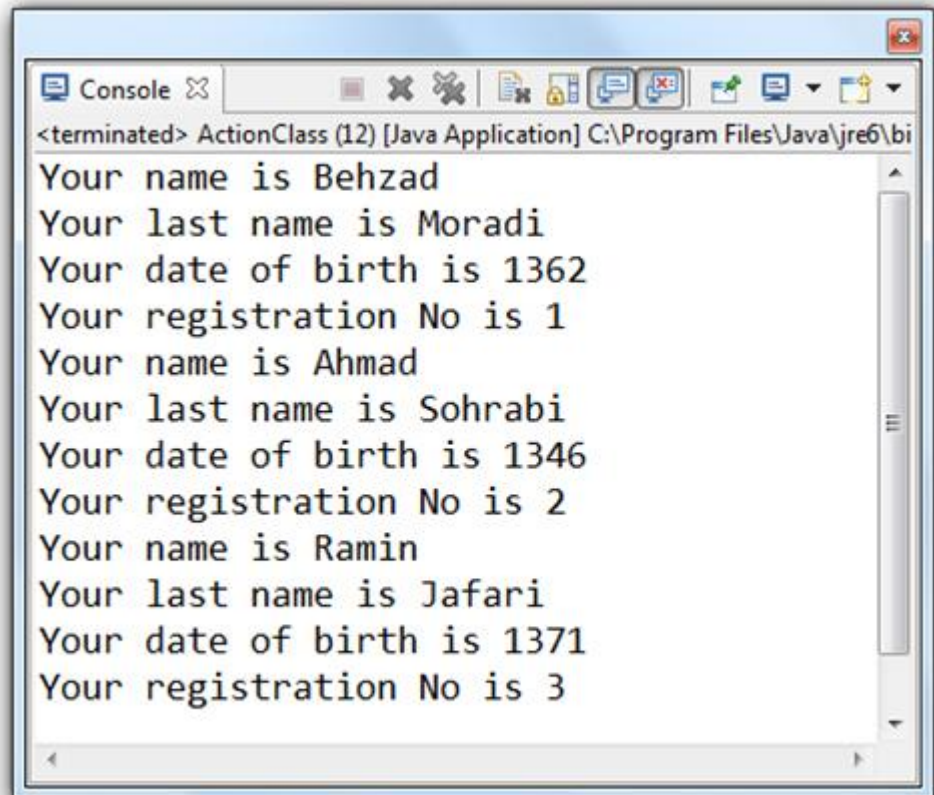


می بینیم که پارامترهای در نظر گرفته شده به نمایش درآمده اما مسئله ای که جالب توجه است این است که شماره عضویت 1 برای Behzad در نظر گرفته شده است و علت آن هم این است که ما مقدار اولیه شماره عضویت را برابر با 0 قرار دادیم و این دستور را به Constructor برنامه دادیم که برای هر شیئی ساخته شده از روی کلاس آن یک واحد به مقدار اولیه اضافه کند و از همین رو است که عدد 1 در اجرای فوق در نظر گرفته شده است.

حال دو شیئی دیگر از روی کلاس Athlete می سازیم:

```
public class ActionClass {  
    public static void main(String[] args) {  
        Athlete personOne = new Athlete("Behzad", "Moradi", 1362);  
        personOne.showAthleteInfo();  
  
        Athlete personTwo = new Athlete("Ahmad", "Sohrabi", 1346);  
        personTwo.showAthleteInfo();  
  
        Athlete personThree = new Athlete("Ramin", "Jafari", 1371);  
        personThree.showAthleteInfo();  
    }  
}
```

همانطور که مشاهده می شود دو شیئی دیگر تحت عناوین personTwo و personThree به معنی به ترتیب "نفر دوم" و "نفر سوم" ایجاد کرده ایم. حال مجدد برنامه را اجرا می کنیم:



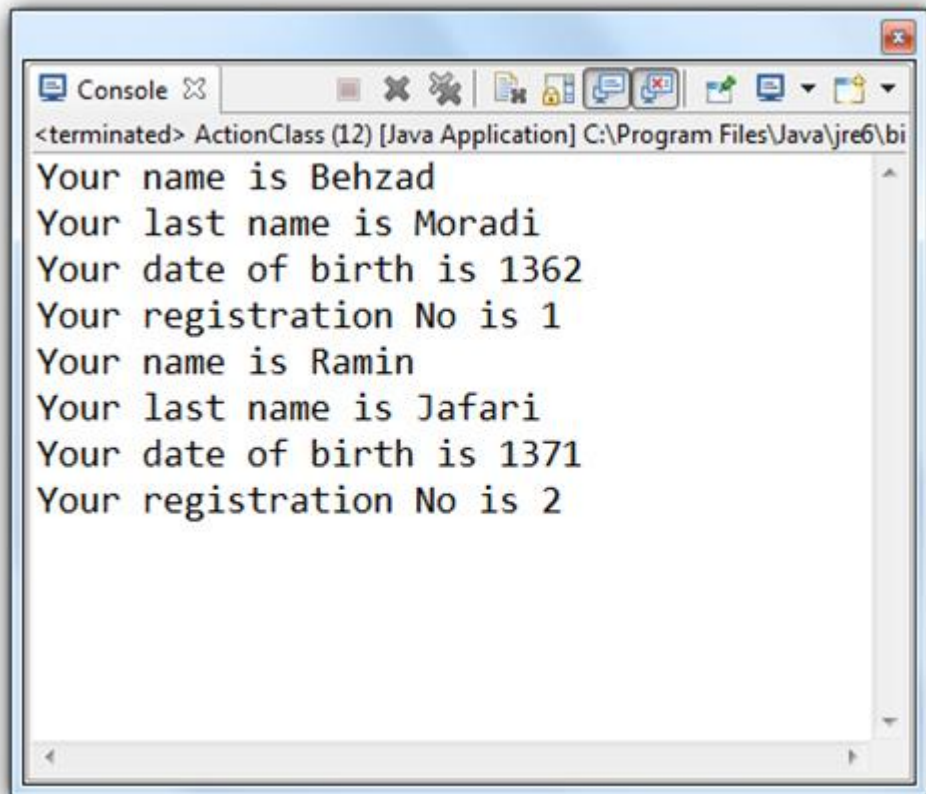
می بینیم که پس از به نمایش درآمدن نام، نام خانوادگی و تاریخ تولد برنامه به صورت خودکار شماره عضویت فرد را اعلام می کند. حال برای تست برنامه `personTwo` را با استفاده از کامنت از کد خود به صورت زیر حذف می کنیم:

```
public class ActionClass {
    public static void main(String[] args) {
        Athlete personOne = new Athlete("Behzad", "Moradi", 1362);
        personOne.showAthleteInfo();

        // Athlete personTwo = new Athlete("Ahmad", "Sohrabi", 1346);
        // personTwo.showAthleteInfo();

        Athlete personThree = new Athlete("Ramin", "Jafari", 1371);
        personThree.showAthleteInfo();
    }
}
```

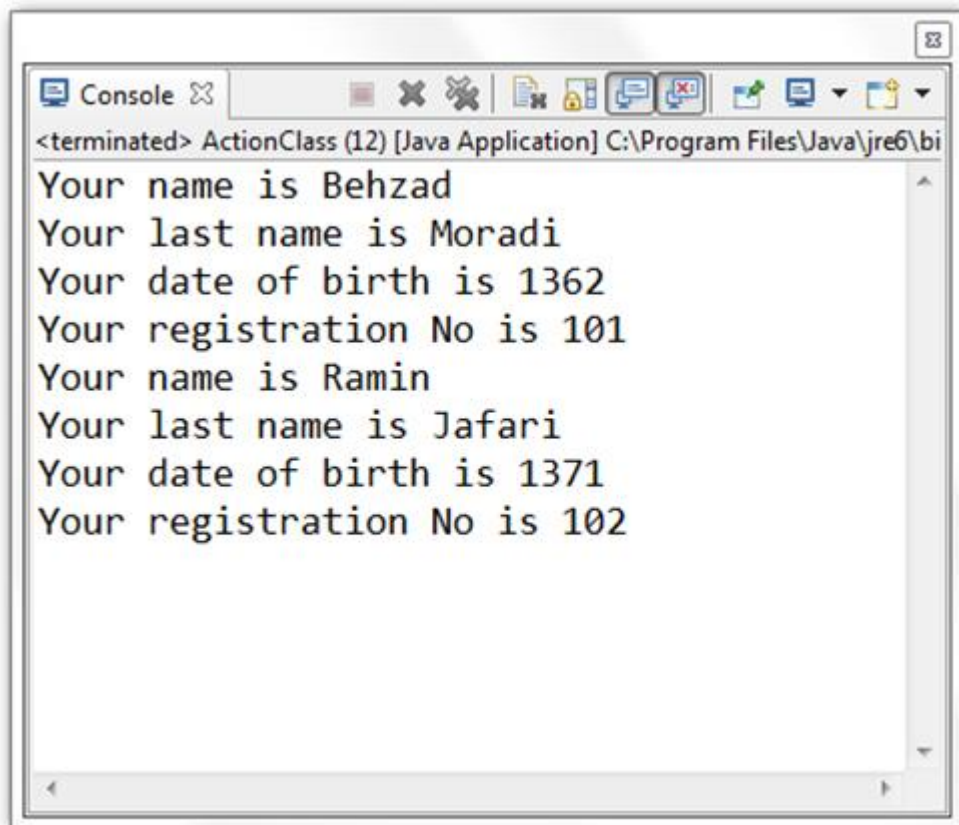
اکنون مجدد برنامه را اجرا می کنیم:



می بینیم که اطلاعات personThree به نمایش درآمده اما این در حالی است که شماره عضویت به صورت خود کار یکی کاهش می یابد. اکنون فرض کنیم که 100 نفر از قبل در باشگاه حضور داشته اند و ما می خواهیم برای نفرات جدید ثبت نامی صورت دهیم. برای این منظور مقدار متغیر شماره عضویت را به صورت زیر به عدد 100 افزایش می دهیم:

```
static int registrationNumber = 100;
```

اکنون مجدد برنامه را اجرا می کنیم:



```
<terminated> ActionClass (12) [Java Application] C:\Program Files\Java\jre6\bin
Your name is Behzad
Your last name is Moradi
Your date of birth is 1362
Your registration No is 101
Your name is Ramin
Your last name is Jafari
Your date of birth is 1371
Your registration No is 102
```

می بینیم که از آنجا که Field مرتبط با شماره عضویت از جنس static است هر تغییری که در آن ایجاد کنیم در کلیه کلاس های ساخته شده از روی آن کلاس اعمال خواهد شد. علاوه بر استفاده کلید واژه static به همراه متغیرها، در زبان برنامه نویسی جاوا این امکان را داریم تا متدها را نیز static سازیم. برای درک بهتر این موضوع مجدد به کلاس Athlete رجوع می کنیم:



```

public class Athlete {
    String name;
    String lastName;
    int dateOfBirth;
    static int registrationNumber = 0;

    public Athlete(String nameOfAthlete, String lastNameOfAthlete,
        int dateOfBirthOfAthlete) {
        name = nameOfAthlete;
        lastName = lastNameOfAthlete;
        dateOfBirth = dateOfBirthOfAthlete;
        registrationNumber++;
    }

    public void showAthleteInfo() {
        System.out.println("Your name is " + name);
        System.out.println("Your last name is " + lastName);
        System.out.println("Your date of birth is " + dateOfBirth);
        System.out.println("Your registration No is " +
registrationNumber);
    }

    public static int showAllAthletes() {
        return registrationNumber;
    }
}

```

همانطور که در کد فوق ملاحظه می شود در انتهای کد اقدام به ساخت یک متد جدید تحت عنوان showAllAthletes به معنی "همه ورزشکاران را نشان بده" نموده ایم. از آنجا که قرار است این متد یک عدد به ما نشان دهد، از این رو آن را از جنس int قرار می دهیم. زمانیکه ما در ساخت یک متد از کلید واژه void استفاده نمی کنیم به بدان معنا است که این متد همواره داده ای را return خواهد کرد پس می بایست کلید واژه return را نوشته و چیزی که قصد داریم این متد بازگرداند را مقابل آن قرار می دهیم. در این مثال می خواهیم این متد همواره تعداد کل ثبت نامی ها را به مدیر باشگاه نشان دهد پس می بایست نام متغیر registrationNumber را مقابل آن بنویسیم.

اکنون مجدد به کلاس ActionClass باز می گردیم. از آنجا که می خواهیم پس از به نمایش در آمدن اطلاعات کلیه شیئی های ساخته شده از روی کلاس Athlete در پنجره Console

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

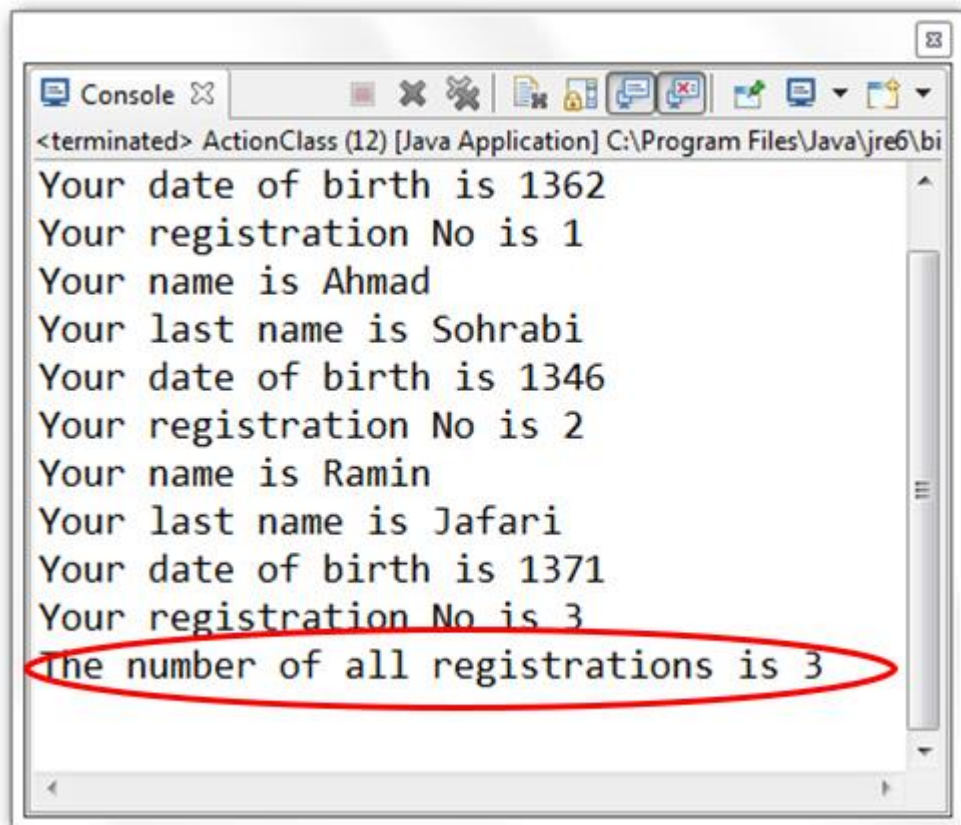
مدرس: بهزاد مرادی



عبارت The number of all registrations is به معنی "تعداد کل ثبت نامی ها معادل است با" روی صفحه به نمایش درآید، از این رو کد خود را به شکل زیر تکمیل می کنیم:

```
public class ActionClass {  
    public static void main(String[] args) {  
        Athlete personOne = new Athlete("Behzad", "Moradi", 1362);  
        personOne.showAthleteInfo();  
  
        Athlete personTwo = new Athlete("Ahmad", "Sohrabi", 1346);  
        personTwo.showAthleteInfo();  
  
        Athlete personThree = new Athlete("Ramin", "Jafari", 1371);  
        personThree.showAthleteInfo();  
  
        System.out.println("The number of all registrations is "  
            + personOne.showAllAthletes());  
    }  
}
```

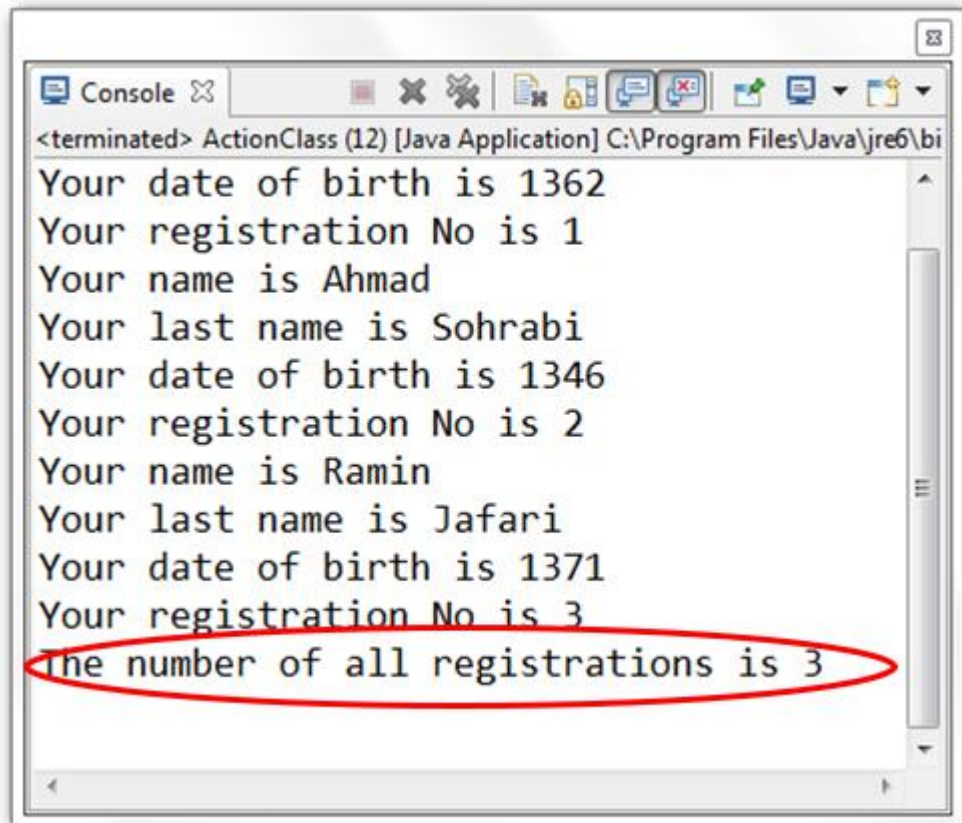
همانطور که در بخش آخر کد مشاهده می کنیم، عبارت مد نظر را داخل دستور System.out.println داخل دو علامت " " نوشته پس از آن یک علامت + قرار داده و در نهایت نام یکی از شیء های ساخته شده از کلاس Athlete را می نویسیم که در این مثال مورد اول که personOne نام داشت را مد نظر قرار داده ایم. پس از نوشتن نام شیء و قرار دادن یک نقطه حال به کلیه متدهای موجود در کلاس اصلی دسترسی خواهیم داشت. از اینرو نام متد showAllAthletes() را می نویسیم. حال برنامه را مجدد اجرا می کنیم:



همانطور که در تصویر فوق با یک بیضی قرمز رنگ نشان داده شده است، تعداد کل ثبت نامی ها به نمایش در آمده است. اکنون برای تست کردن برنامه خود کد فوق را به صورت زیر تغییر می دهیم:

```
System.out.println("The number of all registrations is "  
+ personTwo.showAllAthletes());
```

همانطور که در کد فوق ملاحظه می شود به جای استفاده از شیء اول خود تحت عنوان personOne از شیء دوم ساخته شده از روی کلاس Athlete تحت عنوان personTwo استفاده کرده ایم. حال مجدد برنامه را اجرا می کنیم:



```
<terminated> ActionClass (12) [Java Application] C:\Program Files\Java\jre6\bin
Your date of birth is 1362
Your registration No is 1
Your name is Ahmad
Your last name is Sohrabi
Your date of birth is 1346
Your registration No is 2
Your name is Ramin
Your last name is Jafari
Your date of birth is 1371
Your registration No is 3
The number of all registrations is 3
```

باز هم می بینیم که حاصل متدی که از جنس static بود جدای از شیء مرتبط با آن یکی است و این همان خاصیت کلید واژه static است که این امکان را به ما داده است تا متد showAllAthletes را در میان همه شیء ها اصطلاحاً Share کنیم یا "به اشتراک" بگذاریم. حال برای آنکه مجدد برنامه خود را تست کنیم از قصد یکی از شیء های ساخته شده خود مثلاً شیء سوم خود را مجدد کامنت می کنیم:

```

public class ActionClass {
    public static void main(String[] args) {
        Athlete personOne = new Athlete("Behzad", "Moradi", 1362);
        personOne.showAthleteInfo();

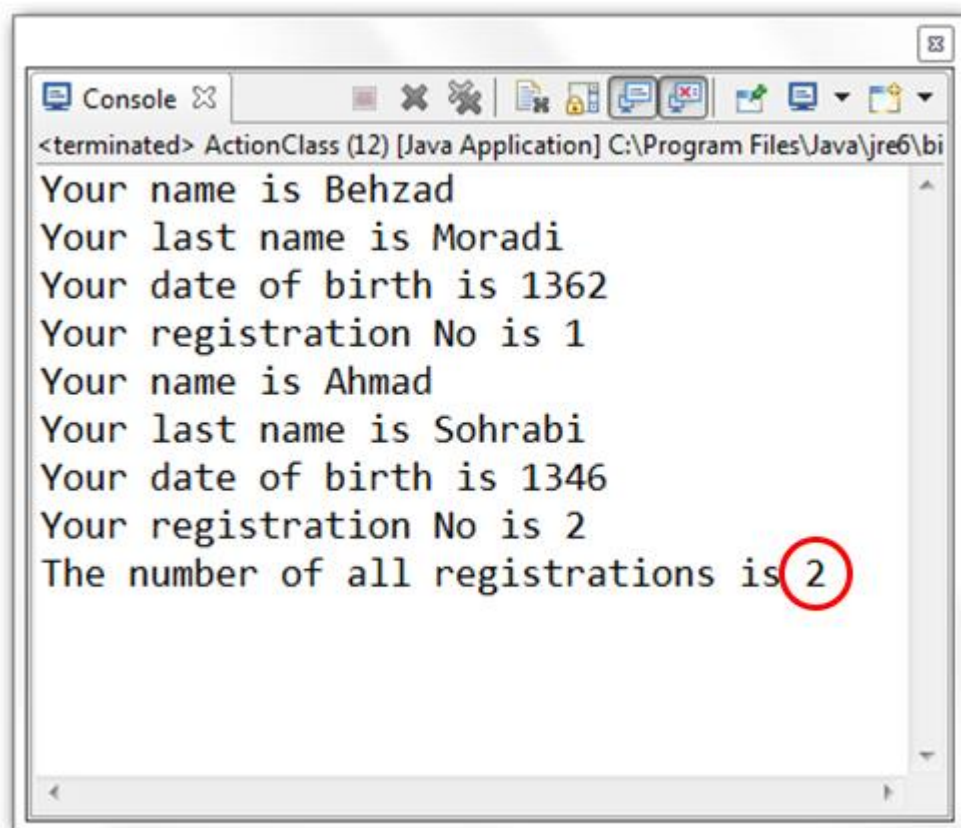
        Athlete personTwo = new Athlete("Ahmad", "Sohrabi", 1346);
        personTwo.showAthleteInfo();

        // Athlete personThree = new Athlete("Ramin", "Jafari", 1371);
        // personThree.showAthleteInfo();

        System.out.println("The number of all registrations is "
            + personOne.showAllAthletes());
    }
}

```

اکنون برنامه را اجرا می کنیم:



می بینیم که برنامه به صورت خودکار تعداد شیء های ساخته شده از روی کلاس Athlete را تشخیص داده و نتیجه آن را روی صفحه به نمایش در می آورد.

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

- پس از مطالعه این آموزش انتظار می رود بتوانیم به سؤالات زیر پاسخ بدهیم:
۱. به چه شکل می توان از روی یک کلاس یک شیء ساخت؟
  ۲. علت استفاده از کلید واژه static در متدها چیست؟