به نام خدا

آموزش چهل و هفتم

اهداف آموزشی این قسمت عبارتند از:

۱. معرفی کلید واژه static در زبان جاوا
۲. به کارگیری از Constructor ها در برنامه های جاوا
۳. استفاده از دستور ++

در این آموزش قصد داریم تا با کلید واژه static در زبان برنامه نویسی جاوا آشنا شویم. معنای لغوی این کلید واژه معادل است با "ثابت" ، "ساکن"و "ایستا" و در حقیقت هر کجای برنامه خود که بخواهیم چیزی همچون یک متغیر را ثابت نگه داریم از این کلید واژه می بایست استفاده نماییم.
در این آموزش برای درک بهتر مفهوم static ابتدا یک سناریو تعریف می کنیم.
فرض کنیم که در یک باشگاه بدنسازی(مخصوص برنامه نویسان آقا) یا یک باشگاه ایروبیک(مخصوص برنامه نویسان خانم) ورزش می کنیم. از آنجا که مدیر این باشگاه می داند که ما یک برنامه نویس هستیم از ما می خواهد که یک برنامه ثبت نام برای برای باشگاه ورزشی بنویسم.
ما هم از این پیشنهاد استقبال کرده و نرم افزار اکلیپس را باز می کنیم!
برای این منظور یک پروژه جدید در محیط اکلیپس تحت عنوان 47^{th} Session ایجاد کرده و کلاسی به نام Athlete به معنی "ورزشکار" در آن ایجاد می کنیم (به خاطر داشته باشیم که برای این کلاس نمی بایست گزینه public static void main را در حین ساخت این کلاس تیک دار نماییم.)
کد ما می بایست به شکل زیر باشد:

```
public class Athlete {

}
```

در حقیقت این کلاس مربوط به بخشی از برنامه ما است که مسئولیت دارد تا اطلاعات تک تک ورزشکاران باشگاه را در خود ذخیره سازد. حال از آنجا که هر متقاضی در باشگاه می بایست نام،

نام خانوادگی، تاریخ تولد و تاریخ ثبت نام خود را در سیستم به ثبت رساند، پس می بایست متغیرهایی ایجاد کنیم که این اطلاعات را در خود ذخیره سازند. برای این منظور کد خود را به شکل زیر تکمیل می کنیم:

```
public class Athlete {
	String name;
	String lastName;
	int dateOfBirth;
	int registrationNumber;
}
```

همانطور که در کد فوق ملاحظه می شود، دو شیئ از روی کلاس String تحت عناوین name و lastName به معنی به ترتیب "نام" و "نام خانوادگی" ایجاد کرده سپس یک متغیر از جنس عدد صحیح تحت عنوان dateOfBirth به معنی "تاریخ تولد" و یک متغیر دیگر هم از همان نوع تحت عنوان registrationNumber به معنی "شماره عضویت" ایجاد کرده ایم.

اکنون ما به هر تعداد ورزشکار که داشته باشیم به همان تعداد می بایست شیئی از روی این کلاس برای آنها ایجاد کنیم و این در حالی خواهد بود که هر کدام از آن شیئ ها دارای name و lastName و dateOfBirth و registrationNumber مخصوص به خود خواهد بود که جایی در حافظه سیستم ذخیره خواهند شد. زمانیکه یک ورزشکار جدید در باشگاه ثبت نام می کند مسلماً دارای نام، نام خانوادگی و تاریخ تولد مختص به خود است اما این در حالی است که شماره عضویت ایشان به هیچ وجه تحت کنترل وی نبوده و به طور مثال نمی تواند برای خود شماره عضویت 7 را در نظر بگیرد. برای این منظور روشی که از آن طریق می توان چنین محدودیتی را ایجاد کرد استفاده از کلید واژه static برای متغیر registrationNumber است. به عبارت دیگر با قرار دادن کلید واژه static پیش از نام یک متغیر این امکان را فراهم خواهیم ساخت که متغیر مد نظر را به متغیری تبدیل کنیم که در همه شیئ های ساخته شده از روی کلاسی که حاوی آن متغیر هستند ثابت باشد. در زبان جاوا به چنین متغیرهایی Static Field گفته می شود. در واقع این دسته از متغیرها مرتبط با شیئ ساخته شده از روی کلاس نبوده بلکه مستقیماً مرتبط با خود کلاس هستند که جای ثابتی را در حافظه سیستم به خود اختصاص می دهند. برای ایجاد چنین محدودیتی کد فوق را به شکل زیر تکمیل می کنیم:

```
public class Athlete {
	String name;
	String lastName;
	int dateOfBirth;
	static int registrationNumber;
}
```

همانطور که در کد فوق ملاحظه می شود پیش از کلید واژه int کلید واژه static را نوشته ایم.

همانطور که در ادامه آموزش خواهیم دید، بخشی از برنامه ما این مسئولیت را خواهد داشت که برای اولین نفر ثبت نامی عدد 1 و برای دهمین نفر ثبت نامی عدد 10 را در نظر خواهد گرفت.

برای این منظور کد فوق را به شکل زیر تکمیل می کنیم:

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

همانطور که در کد فوق ملاحظه می شود مقدار اولیه برای متغیری که از جنس static است معادل با 0 قرار داده شده است و علت این مسئله هم آن است که در حال حاضر هیچ کسی عضو باشگاه نیست.

برای ادامه کار نیاز است تا یک Constructor از روی کلاس خود ایجاد کنیم(برای مطالعه بیشتر پیرامون مفهوم Constructor به آموزش چهل و یکم مراجعه نمایید):

```
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++;
	}
}
```

همانظور که در کد فوق مشاهده می شود یک Constructor ایجاد کرده و برای آن سه پارامتر ورودی تحت عناوین nameOfAthlete و lastNameOfAthlete و dateOfBirthOfAthelte به ترتیب به معانی " نام ورزشکار" و "نام خانوادگی ورزشکار" و "تاریخ تولد ورزشکار" در نظر گرفته ایم. سپس در ادامه کد مربوط به این Constructor مقدار Field خود تحت عنوان name را برابر با nameOfAthlete و مقدار lastName را برابر با lastNameOfAthlete و در نهایت مقدار dateOfBirth را معادل با dateOfBirthOfAthelte قرار داده ایم. به عبارت دیگر کلیه مقادیر مرتبط با پارامترهای شیئ های ساخته شده از روی این کلاس به Field ها یا همان متغیرهای مرتبط با آنها انتقال خواهد یافت. در نهایت با اضافه کردن registrationNumber++ این دستور را به برنامه می دهیم که در هر بار ساخته شدن یک شیئ جدید از روی این کلاس یک واحد به مقدار اولیه این متغیر اضافه کند. در ادامه نیاز داریم تا متدی تعریف کنیم که اطلاعات هر ورزشکار را روی صفحه نمایش دهد. برای این منظور کد فوق را به شکل زیر تکمیل 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);

	}
}
```

همانطور که ملاحظه می شود یک متد از جنس void تحت عنوان showAthleteInfo به معنی "اطلاعات ورزشکار را نشان بده" ایجاد کرده ایم(علت اینکه نوع این متد void انتخاب شده است این می باشد که این متد قرار نیست تا داده ای را return کند). درون این متد چهار بار است دستور System.out.println استفاده کرده ایم تا از آن طریق بتوانیم به ترتیب اطلاعات مربوط به نام، نام خانوادگی، تاریخ تولد و همچنین شماره عضویت ورزشکاران را روی صفحه نمایش دهیم. در آموزش آتی خواهیم دید که به چه نحوه می توان شیئ های دیگری از روی این کلاس ساخت.

پس از مطالعه این آموزش انتظار می رود بتوانیم به سؤالات زیر پاسخ بدهیم:

۱. چرا گاهی اوقات می بایست از دستور static استفاده کرد؟
۲. علت اینکه متد ساخته شده را از جنس void قرار دادیم چه بود؟