

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

وراثت در جاوا

گردآورنده : مهندس افشین رفوآ

وراثت در جاوا

مفهوم مهم دیگر در برنامه نویسی **Object Oriented** (آبجکت محور) مفهوم **Inheritance** می باشد. با چند مثال برنامه نویسی مفهوم **Inheritance** واضح تر خواهد شد. اما لزوما دارای یک گروه به عنوان گروه اصلی می باشد (که گروه **super** نامیده می شود). و یک گروه دیگر به عنوان زیرمجموعه ی گروه اصلی (که گروه **sub** نامیده می شود). گفته می شود که گروه زیرمجموعه از گروه اصلی استخراج شده است. دلیل داشتن یک گروه زیرمجموعه حفظ اطلاعات به صورت مجزا می باشد. گروه زیرمجموعه می تواند تمام متدها و فیلدها را از گروه اصلی خود دریافت کند، اما سپس می تواند کار خود را انجام دهد.

به عنوان یک مثال از **inheritance**، یک گروه زیرمجموعه ایجاد خواهیم کرد که اطلاعات مربوط به تاییدیه ها را بررسی می کند. اگر دانش آموزی رتبه ی "A" دریافت کند، گواهی **(Certificate of Excellence)** به او اعطا خواهیم کرد، اگر دانش آموزی رتبه ی "B" دریافت کند، به او گواهی دستاورد **(Excellence)** به او اعطا خواهیم کرد. برای هر رتبه ی دیگری هیچ گواهی اعطا نمی گردد. اما نکته ی مربوط به گروه زیرمجموعه، نگهداری جدا از هم داده ی مربوط به گواهی و داده ی مربوط به امتحان می باشد. به هر حال ممکن است تمایل داشته باشیم به اطلاعاتی در مورد امتحان دستزسی داشته باشیم، اطلاعاتی از قبیل امتحانی که برگزار شده بود. حتی می توانیم به متدهایی دستزسی داشته باشیم که یک نمره را به رتبه تبدیل می کنند و همه مربوط به زیر مجموعه می باشند.

بنابراین با کلیک کردن بر روی **File > New File** از منوی **NetBeans** یک گروه جدید ایجاد کنید. وقتی که دیالوگ باکس ظاهر می شود، **Java** را در زیر تیتل **Categories** و **Java Class** را زیر **File Types** کلیک کنید. روی **Next** کلیک کرده و **Certificates** را به عنوان نام گروه جدید خود وارد کنید. روی **Finish** کلیک کنید. وقتی که گروه جدید شما ایجاد شده است، یک فیلد خصوصی **String** وارد کرده و آن را **certificate** بنامید. گروه جدید شما باید به شکل زیر باشد.

```
package exams;

public class Certificates {

    private String certificate;

}
```

برای ایجاد یک **sub class** (زیرمجموعه) از گروه اصلی جاوا **super class**، لغت کلیدی **extends** استفاده می شود. بنابراین لغت کلیدی **"extends"** گروه اصلی مورد نظر را که می خواهید گسترش دهید، دنبال می کند. ما می خواهیم که یک گروه زیرمجموعه از **StudentResults** ایجاد کنیم. گروه **StudentResults** گروه اصلی خواهد بود و گروه **Certificates** گروه زیرمجموعه می باشد.

پس از **"public class Certificates"** در کد خود **"extends StudentResults"** را اضافه کنید. بنابراین کد شما باید مانند زیر باشد.

```
package exams;

public class Certificates extends StudentResults {

    private String certificate;

}
```

اکنون شما یک زیرمجموعه دارید که کد را از گروه **StudentResults** دریافت کرده است.

درست مانند گروه **StudentResults**، می توانیم برای این گروه **Certificates** جدید یک سازنده (**constructor**) ایجاد کنیم. وقتی که از گروه یک آبجکت ایجاد می کنیم، اول از همه جاوا سازنده ی ما را فرا می خواند.

به هر حال تنها یک سازنده می تواند فرا خوانده شود. اگر یک سازنده ی جدید را از گروه **Certificates** فرا بخوانیم، همه ی آن مقادیر پیش فرض که برای فیلهای گروه **StudentResults** تنظیم کردیم، تنظیم نخواهند

شد. برای رسیدن به این، یک لغت کلیدی به نام **super** وجود دارد. این لغت از گروه اصلی به سازنده یک فراخوانی می فرستد. سازنده ی زیر را به گروه **Certificates** خود اضافه کنید.

```
package exams;

public class Certificates extends StudentResults {

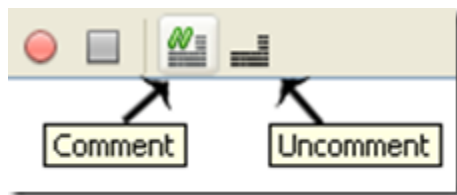
    private String certificate;

    Certificates() {
        super();
        certificate = "No Certificate Awarded";
    }
}
```

نام سازنده هم نام با گروه می باشد: **Certificates**. اولین خط کد بین پراتتزاها گروه اصلی می باشد (به پراتتزاها بعد از **super** توجه کنید). وقتی که این خط اجرا می شود، همه ی فیلدهای پیش فرض برقرار شده در **StudentResults**، تنظیم خواهند شد.

خط دوم کد در سازنده یک مقدار پیش فرض به نام **certificate** برای فیلد **String** تنظیم می کند. (شما در واقع می توانید بیشتر از یک سازنده تنظیم کنید. چگونگی انجام این کار را در بخش های بعد مشاهده خواهید کرد)

برای امتحان کردن گروه جدید خود، به گروه **ExamDetails** بازگردید، گروهی با متد **main**. روی هر کدی که تاکنون داشته اید، کامنت بگذارید. یک راه سریع برای انجام این کار های لایت کردن تمام کد و سپس کلیک کردن روی آیکن **comments** در نوار ابزار **NetBeans** می باشد.



برای رهایی از کامنت ها، دوباره کد را های لایت کرده و روی آیکن **uncomment** کلیک کنید.

اکنون برای ایجاد یک آبجکت جدید از گروه خود، خط زیر را اضافه کنید.

```
Certificates c1 = new Certificates();
```

پنجره ی کد شما باید به شکل زیر باشد.

```
package exams;

public class ExamDetails {

    public static void main(String[] args) {

        //      StudentResults aStudent = new StudentResults();
        //      String sName = aStudent.fullName("Bill Gates");
        //      String exam = aStudent.examName("VB");
        //      String score = aStudent.examScore(30);
        //      String grade = aStudent.examGrade(30);
        //      System.out.println( sName );
        //      System.out.println( exam );
        //      System.out.println( score );
        //      System.out.println( grade );

        Certificates c1 = new Certificates();

    }
}
```

نام آبجکت c1 می باشد و از نوع **Certificates** می باشد.

برای بررسی این موضوع که می توانید به متدهایی از گروه **StudentResults** دسترسی داشته باشید، خط زیر

را به آبجکت جدید **c1** اضافه کنید.

```
String exam = c1.examName("VB");
```

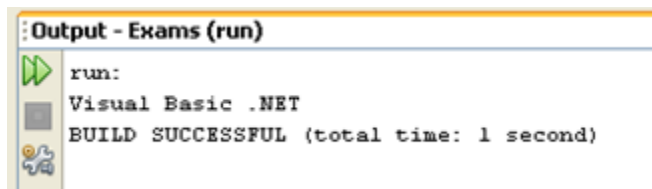
بین درست همان کاریست که قبلاً انجام دادید: فراخوانی متد `examName` از گروه `StudentResults`. به

هرحال این بار از آبجکت `Certificates` به جای آبجکت `StudentResults` استفاده می‌کنید.

یک `print line` اضافه کنید، کد شما باید به شکل زیر باشد.

```
public class ExamDetails {  
  
    public static void main(String[] args) {  
  
        //      StudentResults aStudent = new StudentResults();  
        //      String sName = aStudent.fullName("Bill Gates");  
        //      String exam = aStudent.examName("VB");  
        //      String score = aStudent.examScore(30);  
        //      String grade = aStudent.examGrade(30);  
        //      System.out.println( sName );  
        //      System.out.println( exam );  
        //      System.out.println( score );  
        //      System.out.println( grade );  
  
        Certificates c1 = new Certificates();  
  
        String exam = c1.examName("VB");  
        System.out.println( exam );  
    }  
}
```

برنامه را اجرا کنید تا پنجره ی `Output` را به شکل زیر مشاهده کنید.



بنابراین متد مربوط به گروه اصلی (the super class) به عمل فرا خوانده شده است. اکنون می توانیم یک متد به گروه زیر مجموعه اضافه کنیم (the sub class). متد زیر را به گروه Certificates خود اضافه کنید، درست در زیر constructor.

```
package exams;

public class Certificates extends StudentResults {

    private String certificate;

    Certificates() {
        super();
        certificate = "No Certificate Awarded";
    }

    String certificateAwarded(int aScore) {

        String aGrade = examGrade(aScore);

        if (aGrade.equals("Grade is A") ) {
            this.certificate = "Certificate of Excellence";
        }
        else if (aGrade.equals("Grade is B") ) {
            this.certificate = "Certificate of Achievement";
        }
        else if (aGrade.equals("Grade is C") ) {
            this.certificate = "Certificate of Achievement";
        }
        else {
            this.certificate = "No Certificate Awarded";
        }

        return this.certificate;
    }
}
```

این متد certificateAwarded نامیده می شود و برای بازگرداندن یک مقدار String تنظیم شده است. در داخل پراتتزه‌های مربوط به متد، یک نمونه ی نمره ی آزمون توزیع می کنیم.

اولین خط متد عبارت است از

```
String aGrade = examGrade(aScore);
```

متد **examGrade** یک متد در گروه اصلی می باشد و متدی می باشد که در **StudentResults** تنظیم کردیم. به یاد داشته باشید که این متد برای بازگردانی یک رتبه و عباراتی مانند "Grade is A"، "Grade is B" و غیره تنظیم شده بود. اکنون آن را از زیرمجموعه فرا می خوانیم. **IF Statement** مقدار مربوط به رشته ی **aGrade** را برای مشاهده ی آنچه در آن است، بررسی می کند. بسته به مقدار، یک رشته ی جدید بازگردانده می شود و یک تاییدیه ی مشخص به آن اختصاص داده می شود: **Excellence**، **Achievement** یا بدون تاییدیه. روی **back** برای بازگشت به گروه **ExamDetails** کلیک کنید و خط زیر را اضافه کنید.

```
String award = c1.certificateAwarded(50);
```

این خط متد جدید را فرا خوانده و به آن مقدار 50 اختصاص می دهد. نتیجه به رشته ای که **award** نامیده ایم، بازگردانده می شود.

روش خط چاپ را در کد خود به این روش تطبیق دهید.

```
System.out.println( exam + " " + award );
```

گروه **ExamDetails** باید به شکل زیر باشد. (تمام کامنت ها را حذف کرده ایم)


```

package exams;

public class ExamDetails {

    public static void main(String[] args) {

        Certificates c1 = new Certificates();

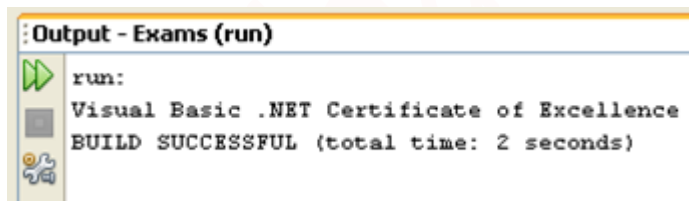
        String exam = c1.examName("VB");
        String award = c1.certificateAwarded(50);

        System.out.println( exam + " " + award );

    }
}

```

و در اینجا پنجره ی **Output** را در هنگام اجرای برنامه مشاهده می کنید.



بنابراین برای نگهداری جزئیات آزمون جدا از جزئیات مربوط به گواهی، از **Inheritance** استفاده کرده ایم. گروه زیرمجموعه به متدهای گروه اصلی خود دسترسی دارد. بنابراین ما قادر به گرفتن خروجی از هر دو هستیم.

بنابراین **Inheritance** نمایش فواید گروه اصلی می باشد. شما می توانید با قرار دادن داده در یک گروه زیرمجموعه، آن را مجزا نگه دارید. ام زیرمجموعه به نحوی در ارتباط با گروه اصلی می باشد و می تواند به تمام یا بخشی از کد آن دسترسی داشته باشد.

در بخش بعد به چگونگی بررسی خطاها در جاوا خواهیم پرداخت.