

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

متدهای بیشتری در جاوا

گردآورنده : مهندس افشین رفوآ

متدهای بیشتری در جاوا

بنابراین ما نام یک دانش آموز و نام یک امتحان داریم. هر دو در فیلد اسم ها در گروه **StudentResults** ذخیره می شوند. اکنون می توانیم یک نمره ی امتحان نیز ذخیره کنیم.

متد جدید زیر را درست بعد از متد **examName** در گروه خود اضافه کنید.

```
String examScore(int aScore){  
  
    Exam_Score = aScore + " out of 50";  
    return Exam_Score;  
}
```

این متد جدید **examScore** نامیده می شود با یک متغیر **int** به نام **aScore** بین پرانتزهای آن. این متد برای بازگرداندن یک مقدار **String** تنظیم شده است. خود متد نمره را با رشته ی "out of 50" ترکیب می کند. بنابراین اگر مقدار در **aScore** عدد 30 باشد، متن "30 out of 50" در فیلد **Exam_Score** ذخیره خواهد شد. در گروه **ExamDetails**، خط زیر را اضافه کنید.

```
String score = aStudent.examScore(30);
```

بنابراین متد جدید **examScore** را فرا می خوانیم و به آن مقدار 30 را می دهیم. مقدار در فیلد **Exam_Score** بازگردانده می شود، و سپس در یک متغیر رشته ذخیره می شود که ما آن را نمره (**score**) نامیده ایم.

یک متد **print line** جدید اضافه کنید، بنابراین کد شما شبیه به کد ما در زیر می شود.

```

public class ExamDetails {

    public static void main(String[] args) {

        StudentResults aStudent = new StudentResults();

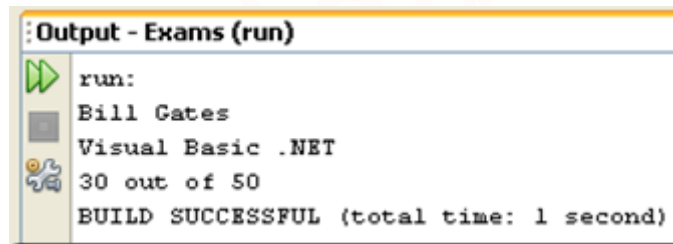
        String sName = aStudent.fullName("Bill Gates");
        String exam = aStudent.examName("VB");
        String score = aStudent.examScore(30);

        System.out.println( sName );
        System.out.println( exam );
        System.out.println( score );

    }
}

```

وقتی برنامه اجرا می شود، پنجره ی **Output** شبیه زیر خواهد بود.



بنابراین ما نام دانش آموز، نام امتحان و نمره از **50** را در دست داریم. اکنون به پنجره ی خروجی می توانیم یک درجه نیز اضافه کنیم.

برای درجه از حروف مجزا استفاده می کنیم: **A, B, C, D**، یا **E**. اگر دانش آموزی نمره ی **41** یا بیشتر بگیرد، حرف **A** را به آن اختصاص می دهیم. اگر نمره بین **31** و **40** باشد، درجه مربوطه **B** خواهد بود. برای نمره ی بین **21** تا **30** حرف **C** اختصاص داده می شود. درجه ی **D** نمره ی بین **11** تا **20** می باشد و حرف **E** برای نمرات بین **0** تا **10** می باشد.

برای محاسبه ی درجات بالا متد زیر را اضافه کنید. (آن را به گروه **StudentResults** خود اضافه کنید)

```

private String getGrade(int aScore) {

    String examGrade = "";

    if (aScore >= 0 && aScore <= 10) {
        examGrade = "E";
    }
    else if (aScore >= 11 && aScore <= 20) {
        examGrade = "D";
    }
    else if (aScore >= 21 && aScore <= 30) {
        examGrade = "C";
    }
    else if (aScore >= 31 && aScore <= 40) {
        examGrade = "B";
    }
    else if (aScore >= 41 ) {
        examGrade = "A";
    }
    return "Grade is " + examGrade;
}

```

دقت داشته باشید که این متد **privat** (خصوصی) می باشد. درست مانند متغیرهای فیلد، خصوصی ساختن یک متد به این معناست که آن متد تنها در داخل این گروه قابل مشاهده می باشد و می تواند به وسیله ی گروه **ExamDetails** مشاهده شود.

برای به دست آوردن درجه، متد دیگری را در داخل گروه **StudentResults** تنظیم خواهیم کرد و از آن برای به دست آوردن درجه استفاده می کنیم. متد زیر را درست در بالای متد **getGrade** اضافه کنید. (گرچه اگر تمایل داشته باشید، می توانید آن را در زیر این متد نیز اضافه کنید: در جاوا هیچ فرقی نمی کند)

```

String examGrade(int aScore) {

    Exam_Grade = this.getGrade( aScore) ;
    return Exam_Grade;
}

```

این متدی است که به جای متد `getGrade` از گروه `ExamDetails` فرا خوانی می کنیم. نام این متد جدید `examGrade` می باشد و مجددا آن را به نمره ی دانش آموز انتقال می دهیم. به این خط دقت کنید.

```
Exam_Grade = this.getGrade( aScore );
```

در اینجا متد `getGrade` فراخوانده می شود و آن را به نمره ای انتقال می دهیم که توزیع شده بود. فراخوانی یک متد از متد دیگر یک تمرین استاندارد می باشد، و به شما اجازه می دهد تا کد خود را ساده کنید. جایگزین آن داشتن متدهای خیلی بلند می باشد که خواندن آنها سخت است.

مورد دیگری که در خط بالا باید به آن توجه داشت لغت کلیدی `this` در جاوا می باشد. لغت کلیدی `this` به معنای `"this class"` می باشد و نه گروه دیگری که ممکن است دارای متد هم نام باشد. این امر از هر گونه سردرگمی جلوگیری می کند. این مسئله زیاد ضروری نیست و می توانید آن را نادیده بگیرید. فراخوانی متد هنوز بدون آن کار می کند.

```
Exam_Grade = getGrade( aScore );
```

گرچه نتیجه ی نهایی هنوز همان است: ما در حال ذخیره سازی چیزی در فیلد متغیر `Exam_Grade` می باشیم و آن عبارت خواهد بود از عبارت `"Grade is"` به علاوه ی یک حرف مربوط به درجه. برای امتحان کردن متدهای جدید، خط زیر را به گروه `ExamDetails` اضافه کنید.

```
String grade = aStudent.examGrade(30);
```

این خط مقداری برابر 30 را به متد `examGrade` می دهد. سپس در فیلد متغیر `Exam_Grade` مقداری گزارش می شود و در متغیری به نام `grade` ذخیره می شود. با یک `print line`، گروه `ExamDetails` باید به شکل زیر باشد.

```

public class ExamDetails {

    public static void main(String[] args) {

        StudentResults aStudent = new StudentResults();

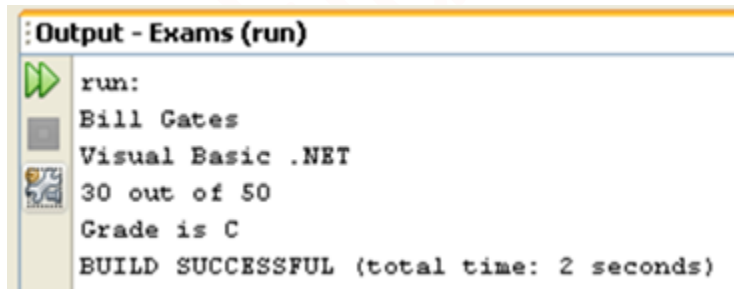
        String sName = aStudent.fullName("Bill Gates");
        String exam = aStudent.examName("VB");
        String score = aStudent.examScore(30);
        String grade = aStudent.examGrade(30);

        System.out.println( sName );
        System.out.println( exam );
        System.out.println( score );
        System.out.println( grade );

    }
}

```

برنامه ی خود را اجرا کنید تا پنجره ی **Output** را مشاهده کنید.



اگر برنامه ی شما به درستی کار نمی کند، در اینجا کد کامل مربوط به گروه **StudentResults** را مشاهده می کنید.

```

package exams;

public class StudentResults {

    private String Full_Name;
    private String Exam_Name;
    private String Exam_Score;
    private String Exam_Grade;

    StudentResults() {
        Full_Name = "No Name Given";
        Exam_Name = "Unknown";
        Exam_Score = "No Score";
        Exam_Grade = "Unknown";
    }

    String fullName(String aName) {

        Full_Name = aName;
        return Full_Name;
    }

    String examName(String examCode) {

        if (examCode.equals("VB")) {
            Exam_Name = "Visual Basic .NET";
        }
        else if (examCode.equals("JV")) {
            Exam_Name = "Java";
        }
        else if (examCode.equals("C#")) {
            Exam_Name = "C# .NET";
        }
        else if (examCode.equals("PH")) {
            Exam_Name = "PHP";
        }
        else {
            Exam_Name = "No Exam Selected";
        }

        return Exam_Name;
    }
}

```

```

String examScore(int aScore) {

    Exam_Score = aScore + " out of 50";
    return Exam_Score;
}

String examGrade(int aScore) {
    Exam_Grade = this.getGrade(aScore);
    return Exam_Grade;
}

private String getGrade(int aScore) {

    String examGrade = "";

    if (aScore >= 0 && aScore <= 10) {
        examGrade = "E";
    }
    else if (aScore >= 11 && aScore <= 20) {
        examGrade = "D";
    }
    else if (aScore >= 21 && aScore <= 30) {
        examGrade = "C";
    }
    else if (aScore >= 31 && aScore <= 40) {
        examGrade = "B";
    }
    else if (aScore >= 41 ) {
        examGrade = "A";
    }
    return "Grade is " + examGrade;
}
}

```

در بخش بعد در مورد **Inheritance** فرا خواهید گرفت.