

به نام خدا

آموزش سی ام

اهداف آموزشی این قسمت عبارتند از:

۱. نحوه جمع کردن Element های یک Array با یکدیگر

۲. آشنایی با Enhanced for Loop

گاهی اوقات در طراحی یک برنامه نیاز است تا عناصر تشکیل دهنده یک Array را با یکدیگر جمع کنیم. برای درک این مسئله یک سناریوی جدید تعریف می کنیم. فرض کنیم که می خواهیم برنامه ای بنویسیم که در آن بتوانیم میانگین درآمد ماهیانه سال گذشته خود را مشخص کنیم (البته برای نوشتن چنین برنامه ای راه های بسیاری وجود دارد که حتی ممکن است به مراتب از استفاده از Array ها نیز راحت تر باشد. اما در این آموزش صرفاً هدف درک نحوه کارکرد Array ها می باشد).

برای این منظور ابتدا پروژه ای تحت عنوان 30th Session به معنی "جلسه سی ام" ایجاد کرده، سپس یک کلاس جدید تحت عنوان MonthlyPaymentAverage به معنی "میانگین درآمد ماهیانه" ایجاد می کنیم. اکنون یک Array از جنس int ایجاد کرده و کلیه ماه های سال را برای آن تعریف می کنیم. در این مرحله کد ما می بایست به شکل زیر باشد:

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

```

public class MonthlyPaymentAverage {
    public static void main(String[] args) {
        int[] monthNames = new int[12];
        monthNames[0] = 500000;
        monthNames[1] = 1500000;
        monthNames[2] = 1300000;
        monthNames[3] = 1700000;
        monthNames[4] = 2500000;
        monthNames[5] = 500000;
        monthNames[6] = 1500000;
        monthNames[7] = 1000000;
        monthNames[8] = 750000;
        monthNames[9] = 500000;
        monthNames[10] = 1400000;
        monthNames[11] = 1100000;
    }
}

```

در کد فوق تعریف کرده ایم که در Array ما دوازده Element جای گیرد. سپس برای هر یک از ماه ها یک درآمد ماهیانه در نظر گرفته ایم (اگر توجه کرده باشیم می بینیم که اولین ماه که فروردین باشد جایگاه صفر را به خود اختصاص داده است).

حال یک متغیر از جنس int ایجاد می کنیم و نام آن را total به معنی "کل" می گذاریم. مقدار در نظر گرفته شده برای این متغیر صفر خواهد بود. سپس یک Loop از جنس for ایجاد می کنیم چرا که می خواهیم از طریق Loop در هر دوازده ماه سال گردش کرده تا مقدار آن ها را گرفته سپس میانگین آن ها را محاسبه کنیم. یک متغیر از جنس int داخل for ایجاد کرده و نام آن را counter به معنی "شمارنده" می گذاریم و Value اولیه آن را صفر می گذاریم چرا که اولین گزینه در Array یی که در بالا ایجاد شده است از صفر شروع می شود. به عبارت دیگر int counter = 0; به منزله نقطه آغازین می باشد. سپس نقطه پایانی را کوچکتر از عدد ۱۲ می گذاریم زیرا از آن جا که ما ۱۲ ماه در سال داریم پس می خواهیم که این Loop یازده بار گردش کند و در نهایت به for دستور می دهیم که در هر بار Loop شدن یک واحد به متغیر counter اضافه کند. در این مرحله کد ما می بایست به شکل زیر باشد (علت آنکه به Loop خود دستور می دهیم که یازده بار گردش کند این است که یازده بار برای ماه های یک تا یازده به علاوه یک بار برای ماه شماره صفر):

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

```

public class MonthlyPaymentAverage {
    public static void main(String[] args) {
        int[] monthNames = new int[12];
        monthNames[0] = 500000;
        monthNames[1] = 1500000;
        monthNames[2] = 1300000;
        monthNames[3] = 1700000;
        monthNames[4] = 2500000;
        monthNames[5] = 500000;
        monthNames[6] = 1500000;
        monthNames[7] = 1000000;
        monthNames[8] = 750000;
        monthNames[9] = 500000;
        monthNames[10] = 1400000;
        monthNames[11] = 1100000;

        int total = 0;
        for (int counter = 0; counter < 12; counter++) {
            }
        }
    }
}

```

حال می خواهیم به Loop خود دستور می دهیم که در هر بار Loop شدن مقدار یکی از ماه های سال را گرفته و آن را به متغیر total اختصاص دهد. برای این منظور نام متغیر total را نوشته و یک علامت مساوی مقابل آن قرار می دهیم. در واقع با این کار می خواهیم مقدار جدید را به این متغیر اختصاص دهیم. حال این مقدار جدید monthNames خواهد بود که مقابل آن یک علامت [] نیز قرار گرفته است. در واقع عددی که داخل علامت [] قرار می گیرد حاکی از ماهی است که قرار است مقدار اختصاص داده شده به آن به متغیر total اختصاص یابد. برای اینکه این کار به صورت خود کار صورت گیرد داخل علامت [] به جای این که یک عدد از صفر تا یازده بنویسیم که کل ماه ها را شامل شود، نام متغیری که در for ایجاد کردیم را خواهیم نوشت چرا این متغیر نیز از صفر شروع شده و یازده بار نیز تکرار خواهد شد. کد ما به شکل زیر تکمیل خواهد شد:

```

public class MonthlyPaymentAverage {
    public static void main(String[] args) {
        int[] monthNames = new int[12];
        monthNames[0] = 500000;
        monthNames[1] = 1500000;
        monthNames[2] = 1300000;
        monthNames[3] = 1700000;
        monthNames[4] = 2500000;
        monthNames[5] = 500000;
        monthNames[6] = 1500000;
        monthNames[7] = 1000000;
        monthNames[8] = 750000;
        monthNames[9] = 500000;
        monthNames[10] = 1400000;
        monthNames[11] = 1100000;

        int total = 0;
        for (int counter = 0; counter < 12; counter++) {
            total = monthNames[counter];
        }
    }
}

```

نکته ای که در اینجا می بایست مد نظر قرار داده شود این است که عبارت `total = monthNames[counter];` داده شده به متغیر `total` را معادل با مقدار `Array` قرار دهد. اما چیزی که ما از برنامه خود می خواهیم این است که این مقادیر با یکدیگر جمع شوند. از این رو یک علامت `+` قبل از علامت مساوی قرار می دهیم. در واقع با این کار در هر بار `Loop` شدن مقدار اختصاص داده شده به همه `Element` های موجود در `Array` به متغیر `total` اضافه می شود. حال برای آنکه خروجی برنامه روی صفحه مانیتور به نمایش در آید، کد خود را به شکل زیر تکمیل می کنیم:

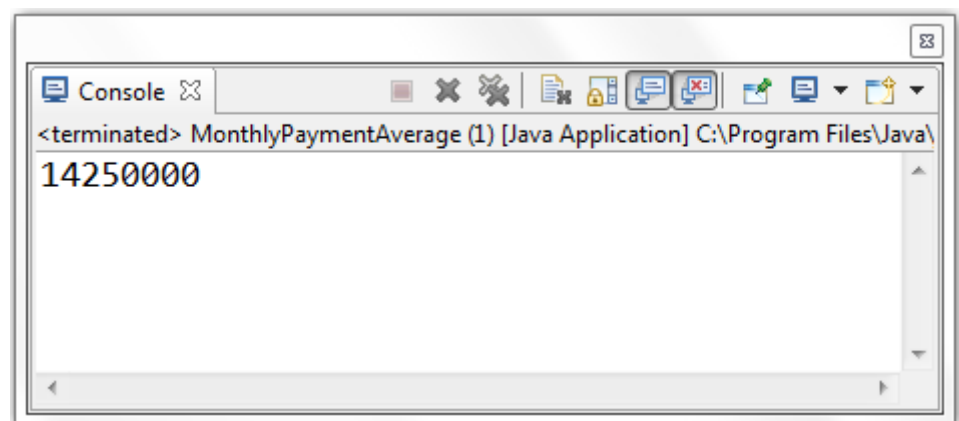
```

public class MonthlyPaymentAverage {
    public static void main(String[] args) {
        int[] monthNames = new int[12];
        monthNames[0] = 500000;
        monthNames[1] = 1500000;
        monthNames[2] = 1300000;
        monthNames[3] = 1700000;
        monthNames[4] = 2500000;
        monthNames[5] = 500000;
        monthNames[6] = 1500000;
        monthNames[7] = 1000000;
        monthNames[8] = 750000;
        monthNames[9] = 500000;
        monthNames[10] = 1400000;
        monthNames[11] = 1100000;

        int total = 0;
        for (int counter = 0; counter < 12; counter++) {
            total += monthNames[counter];
        }
        System.out.println(total);
    }
}

```

با استفاده از دستور `System.out.println()` و قرار دادن نام متغیر `total` در آن، جمع کل درآمدهای یک سال روی صفحه مانیتور به نمایش در خواهد آمد. خروجی برنامه به شکل زیر خواهد بود:



حال نیاز داریم تا این عدد را بر عدد ۱۲ تقسیم کنیم تا میانگین درآمد هر ماه محاسبه شود. برای این منظور کد خود را به شکل زیر تکمیل می کنیم:

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

```

public class MonthlyPaymentAverage {
    public static void main(String[] args) {
        int[] monthNames = new int[12];
        monthNames[0] = 500000;
        monthNames[1] = 1500000;
        monthNames[2] = 1300000;
        monthNames[3] = 1700000;
        monthNames[4] = 2500000;
        monthNames[5] = 500000;
        monthNames[6] = 1500000;
        monthNames[7] = 1000000;
        monthNames[8] = 750000;
        monthNames[9] = 500000;
        monthNames[10] = 1400000;
        monthNames[11] = 1100000;

        int total = 0;
        for (int counter = 0; counter < 12; counter++) {
            total += monthNames[counter];
        }
        int averageIncome = total / 12;
        System.out.println("Your average income is " + averageIncome);
    }
}

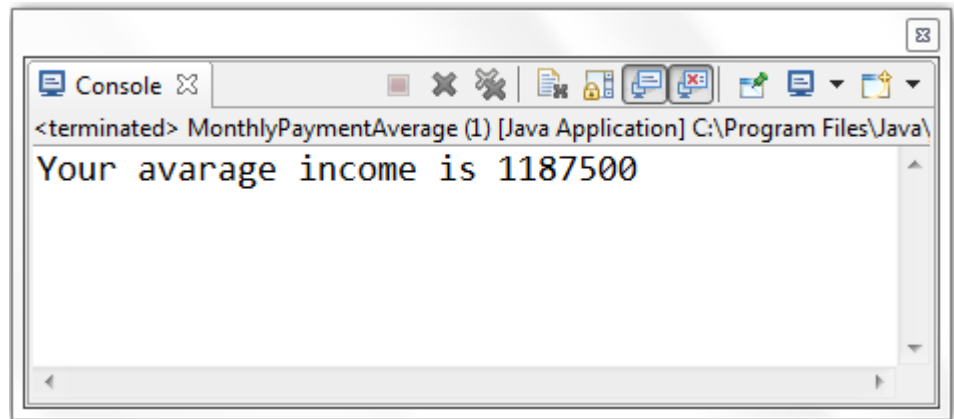
```

در کد فوق یک متغیر از جنس `int` به نام `averageIncome` به معنای "میانگین درآمد" تعریف کرده ایم و مقدار آن را معادل با مقدار متغیر `total` تقسیم بر تعداد ماه های یک سال قرار داده ایم. سپس در دستور `System.out.println()` عبارت `Your average income is` به معنای "میانگین درآمد ماهیانه شما معادل است با" را نوشته و پس از قرار دادن یک علامت + نام متغیری که میانگین درآمد ماهیانه را محاسبه کرده است را می نویسیم. پس از اجرای برنامه خروجی زیر مشاهده خواهد شد:

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی



اگر خاطرمان باشد در آموزش بیست و دوم با Loop یی از جنس for آشنا شدیم که این نوع Loop به منزله شکل ساده ای از for می باشد. اکنون که با مفهوم Array ها آشنا شده ایم به راحتی می توانیم شکل جدیدی از for ها را تحت عنوان Enhanced for Loop را مورد بررسی قرار دهیم (ترجیح این است که کمتر دست به ترجمه اصطلاحات تخصصی برنامه نویسی بزنیم اما برای روشن تر شدن تفاوت مابین for Loop و Enhanced for Loop می توان مورد دوم را به "یک Loop پیشرفته از جنس for" ترجمه کرد).

به طور کلی می توان گفت که Enhanced for Loop در جاوای نسخه 5.0 معرفی شد و هدف از طراحی آن ساده سازی به کار گیری for بود. از سوی دیگر با توجه به مثال به کار گرفته شده در این آموزش، اگر ما از Enhanced for Loop استفاده می کردیم دیگر نیازی به به کار گیری از سومین پارامتر داخل for که همان مقدار افزایشی در هر بار Loop شدن است نخواهد بود. برای روشن تر شدن این مطلب کلاس جدیدی تحت عنوان EnhancedForLoop در پروژه فعلی خود می سازیم. حال می بایست کلاسی به شکل زیر داشته باشیم:

```
public class EnhancedForLoop {  
    public static void main(String[] args) {  
    }  
}
```

اکنون نیاز است تا یک Array در کلاس خود 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

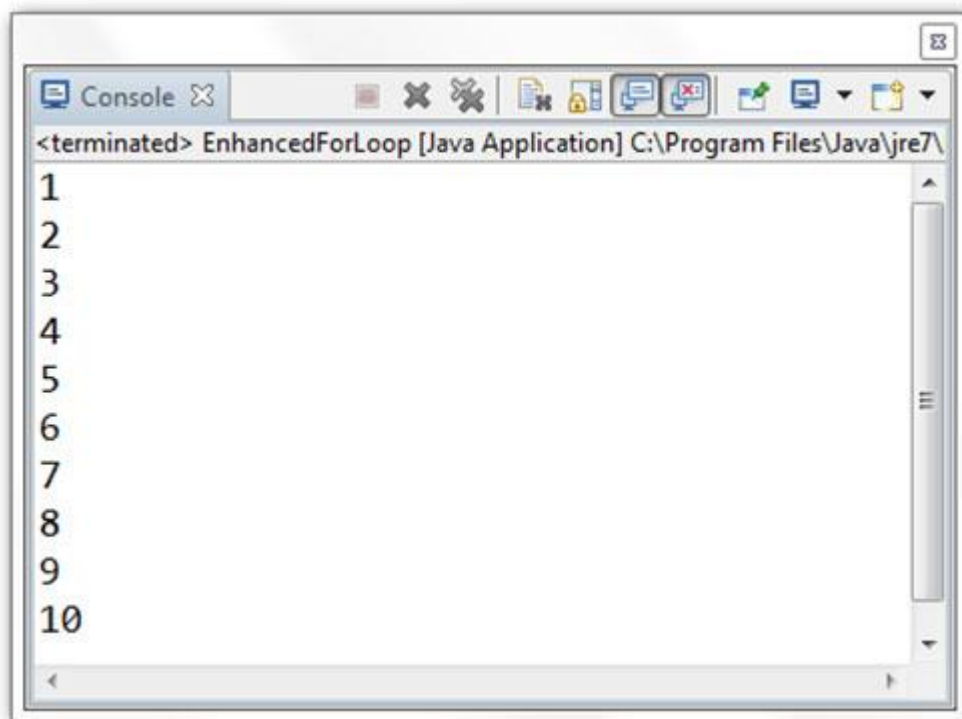
```
public class EnhancedForLoop {

    public static void main(String[] args) {
        int numbers[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    }
}
```

همانطور که ملاحظه می شود یک Array از جنس عدد صحیح تحت عنوان numbers به معنی "اعداد" ساخته و مقدار آن را اعداد 1 الی 10 قرار داده ایم. حال قصد داریم تا با استفاده از یک Enhanced for Loop مقادیر این Array را یکی پس از دیگری نشان دهیم. برای این منظور کد فوق را به شکل زیر تکمیل می کنیم:

```
public class EnhancedForLoop {
    public static void main(String[] args) {
        int numbers[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
        for (int total : numbers) {
            System.out.println(total);
        }
    }
}
```

به طور خلاصه می توان گفت که ساختار یک Enhanced for Loop همانند یک for عادی است با این تفاوت که به جای آنکه سه پارامتر را داخل پرانتز for بنویسیم به یک پارامتر اکتفا کرده سپس بعد از آن یک علامت : قرار داده و در نهایت نام Array یی که می خواهیم Loop شود را می نویسیم. در مثال فوق پارامتر در نظر گرفته شده برای for ما یک متغیر از جنس int تحت عنوان total به معنی "کل" است که مقدار آن برابر با نام Array یی است تحت عنوان numbers که پیش از این ایجاد کردیم. حال برنامه را اجرا می کنیم:



همانطور که در تصویر فوق ملاحظه می شود به سهولت توانستیم اقدام به ایجاد یک Loop پیشرفته با نوشتن کد کمتری نماییم.

در آموزش سی و یکم با نحوه ایجاد دو کلاس متفاوت و ساخت یک Object از روی یکی از کلاس ها و فرا خواندن آن در کلاسی دیگر آشنا خواهیم شد. در واقع در این آموزش اولین شیئی خود در یک زبان برنامه نویسی شیئی گرا همچون جاوا را ایجاد خواهیم کرد. پس از مطالعه این آموزش انتظار می رود بتوانیم به سؤالات زیر پاسخ بدهیم:

۱. چرا زمانیکه یک Array ایجاد کردیم عدد 12 را برای آن در نظر گرفتیم ولی زمانیکه اقدام به نوشتن Element های آن Array کردیم آخرین جایگاه به عدد 11 اختصاص داده شد؟

۲. منظور از Enhanced for Loop در زبان برنامه نویسی جاوا چیست؟

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی