کنترل اجرا

پیش نیاز-آرایه

- آرایه ساختمان داده ای است که از مجموعه ثابتی از اعداد با عناصر داده ای همگن ایجاد شده است.
- سایز هر آرایه ثابت بوده و قابل تغییر نمی باشد.
- در زبان برنامه نویسی جاوا، آرایه ها نوعی object به حساب می آیند.آرایه ها می توانند از نوع داده های اولیه و یا نوع reference باشند.
- هر شی آرایه دارای یک فیلد نهایی به نام طول می باشد که سایز آرایه را مشخص می کند.

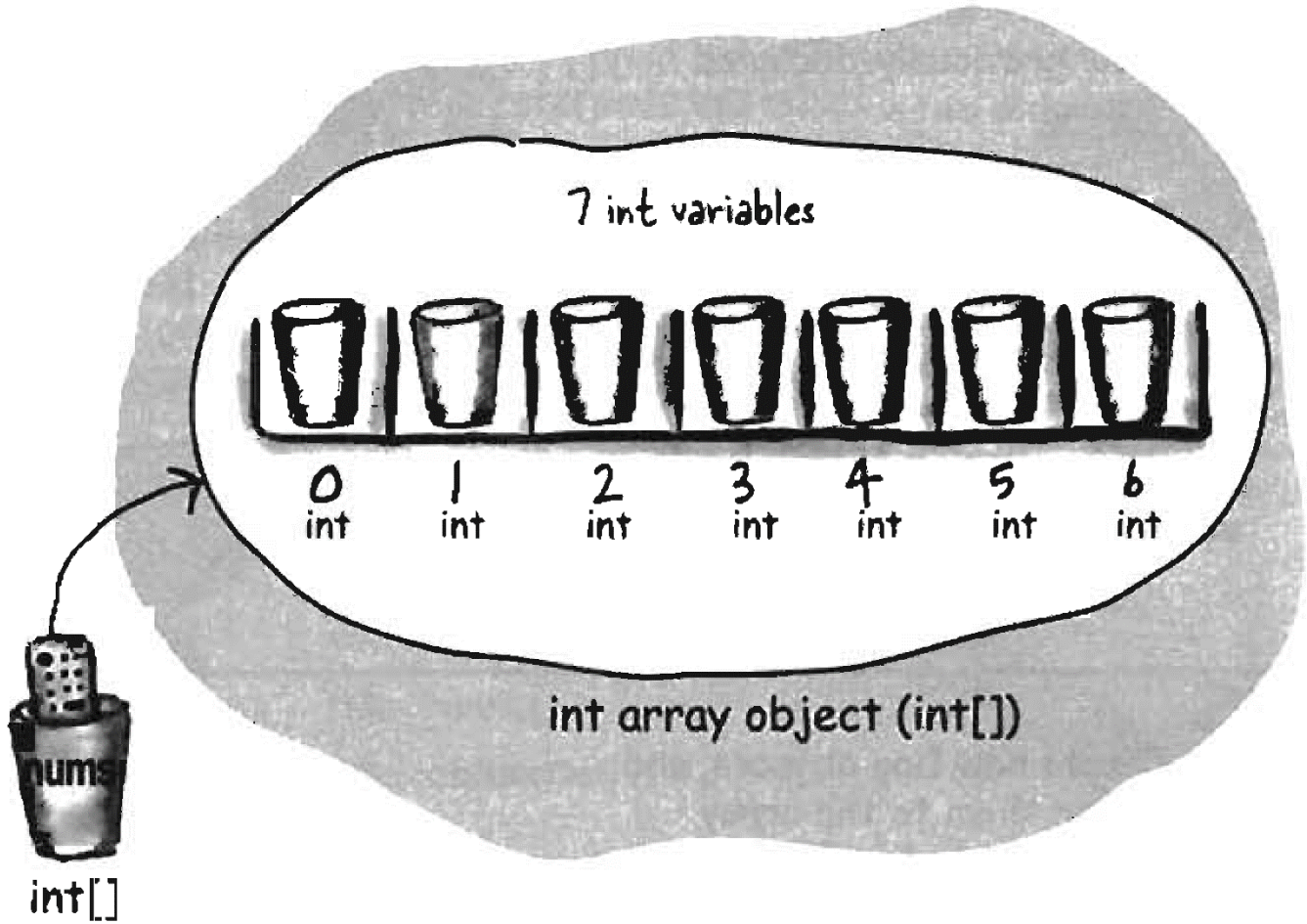
```
int[] anIntArray = new int[10];
Pizza[] mediumPizzas = new Pizza[5];
```

آرایه

```
1  int[] nums;

2  nums = new int[7];

3  nums[0] = 6;
   nums[1] = 19;
   nums[2] = 44;
   nums[3] = 42;
   nums[4] = 10;
   nums[5] = 20;
   nums[6] = 1;
```



کنترل اجرا

- یک برنامه همانند موجودی حساس باید جهان خود را تحت تاثیر قرار داده و در طول اجرا اختیاراتی را ایجاد کند.
- جاوا از تمام دستورات کنترلی زمان اجرای موجود در زبان c استفاده می کند و در صورتی که با زبان c و یا c++ برنامه نویسی کرده باشید بیشتر عباراتی که خواهید دید آشنا خواهند بود.
- در جاوا کلید واژه های break، return، for، do-while، while، if-else و عبارات انتخابی که switch نامیده می شوند،موجود است.

if-else

- دستورات if-else اساسی ترین راه برای کنترل جریان برنامه هستند.قسمت else اختیاری است.

```
if(Boolean-expression)
        statement
else
    statement
```

حلقه تکرار while

دستوری که تا زمانی که عبارت کنترلی boolean موجود false نشده،تکرار می شود.
فرم حلقه while به صورت زیر است:

```
while(Boolean-expression)
    statement
```

حلقه تکرار for

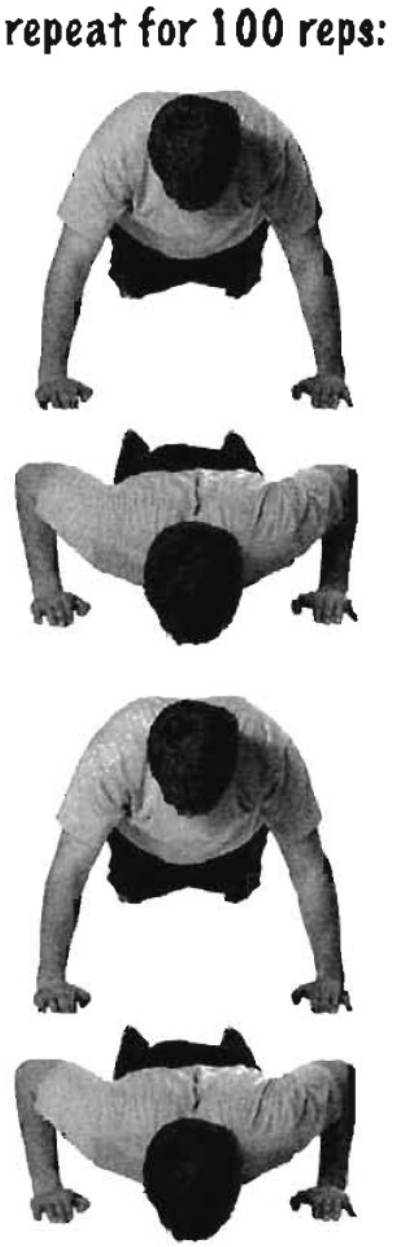
- حلقه تکرار for متداول ترین شیوه استفاده از دستورات تکرار است.در این حلقه،پیش از شروع دستورات حلقه ابتدا یک مقداردهی اولیه صورت می گیرد.سپس بررسی شرط حلقه و در انتهای هر حلقه نوعی مرحله بندی صورت می گیرد.

```
for(initialization; Boolean-expression; step)   statement
```

- هر کدام از قسمت های عبارات اولیه،عبارات بررسی و boolean و مرحله بندی، می توانند خالی باشند.

```
for(char c = 0; c < 128; c++)
    if(Character.isLowerCase(c))
        System.out.println("value: "+(int)c +" character: "+c);
```

حلقه تکرار for




```
for(int i = 0; i < 100; i++) { }
```

initialization

boolean test

post-increment operator

iteration expression

the code to repeat goes here (the body)

اپراتور کاما

- در دو قسمت مقداردهی اولیه و مرحله بندی موجود در دستورات کنترلی، می توان تعدادی از عبارات را که از طریق کاما از هم جدا می شوند،در نظر گرفت.این عبارات به ترتیب محاسبه خواهند شد.

```
for(int i = 1, j = i + 10;
    i < 5;
    i++, j = i * 2)
```

دستور تکرار foreach

- جاوا SE5 نوع جدید و مختصری از دستور for را ایجاد کرده است که با آرایه ها و container ها کار می کند و به آن دستور foreach می گویند.با استفاده از این دستور دیگر احتیاجی به ایجاد یک شمارنده صحیح برای ایجاد توالی آیتم ها نیست و دستور foreach هر آیتم را به صورت خودکار ایجاد می کند.

```
float f[] = new float[10];
//initial f
for(float x : f)
    System.out.println(x);
```

break and **continue**

- برخی دستورات هستند که اجرای بدون شرط قسمتی را فراهم می کنند.این دستورات عبارتند از break، return و continue (و شیوه ای برای پرش به دستورات برچسب زده شده)
  - ✓ دستور break بدون اجرای بقیه دستورات موجود در حلقه،اجرا را از حلقه خارج می کند.
  - ✓ دستور continue اجرای تکرار فعلی را متوقف نموده و برای اجرای تکرار بعدی به ابتدای حلقه می رود.
  - ✓ دستور return از حلقه و متد به همراه مقداری خارج می شود.

break and **continue**

```
int i = 0;
while(true) {
    i++;
    int j = i * 27;
    if(j == 1269) break;
    if(i % 10 != 0) continue;
    System.out.print(i + " ");
}
```

- حلقه بی نهایت while(true) در تئوری می تواند تا بی نهایت ادامه یابد.
- فرم دوم حلقه بی نهایت به صورت (;;) for می باشد.

دستور نامتداول goto

- عبارت goto از ابتدا در زبان های برنامه نویسی موجود بوده است.“اگر شرایط A موجود بود به این دستور پرش کن و در غیر این صورت به مکان مشخص دیگری پرش کن“
- زبان جاوا دستوری تحت عنوان goto ندارد. گرچه گاهی دستوراتی دارد که از عبارات break و continueاستفاده نموده و کمی شبیه به goto عمل می کنند.
- البته این یک پرش نیست و تنها راهی برای پایان دادن و خروج از یک دستور تکرار می باشد.

```
Label1:
```

دستور نامتداول goto

- در زبان جاوا تنها مکانی که استفاده از label مفید است درست قبل از شروع یک عبارت تکرار است.

```
label1:
  outer-iteration {
    inner-iteration {
    //...
    break; // (1)
    //...
    continue; // (2)
    //...
    continue label1; // (3)
    //...
    break label1; // (4)
  }
}
```

switch

- دستورات switch نوعی از دستورات کنترل جریان است که با یک عبارت شروع شده و بسته به مقدار عبارت، کنترل به یکی از دستورات case منتقل می شود.switch با int، short، byte وchar،همچنین نوع Wrapper Type ها Short، Byte، character و Integer،نوع شمارشی enum و همینطور با رشته ها کار می کند.
- توجه داشته باشید که هر case با یک break پایان می یابد و پس از اجرای دستورات case مورد نظر،اجرا به پایان دستور switch انتقال می یابد.

switch

```
/* local variable definition */
char grade = 'B';
String result = null;

switch(grade)
{
case 'A' :
   result = "Excellent!";
   break;
case 'B' :
case 'C' :
   result = "Well done";
   break;
case 'D' :
   result = "You passed";
   break;
case 'F' :
   result = "Better try again";
   break;
default :
   result = "Invalid grade";
}
```