

JAVA DÖNGÜ DEYİMLERİ

Belirli bir iş bir çok kez tekrarlanacaksa, programda bu iş bir kez yazılır ve döngü deyimleriyle istenildiği kadar tekrarlanabilir. Java'da bu işi yapan üç ayrı deyim vardır: while döngüsü, do..while .. döngüsü, for döngüsü.

For Döngüsü

For döngüsü, belirli bir sayıda yürütülmesi gereken döngüyü, verimli bir şekilde yazmanıza olanak sağlayan bir tekrarlı kontrol yapısıdır. For döngüsü, işlemlerin tekrar sayısının önceden belli olduğu durumlarda yararlıdır. Diğer bir genel kullanımda bir dizinin elemanlarını sıralarken kullanılmasıdır. Kısaca for döngüsünün yapısı aşağıdaki gibidir.

```
[crayon-5c3f43b90cce9511072718/]
```

For yapısını daha detaylı inleyecek olursak:

İlk parça, başlangıç değeri (initialization) dir. Burada bulunan işlemler, döngü başlamadan önce ve bir kere işlenir. Bu 1. değere birden fazla işlem yazılabilir, bunları da birbirinden ',' virgül ile ayırmak gerekir. Ancak bu işlemlerin az tutulması daha iyi olacaktır.

İkinci parçada, koşul kontrolü yapılır. For döngüsünün her tekrarında kontrol edilmesi gereken kodu bu kısma yazmak gerekir. Bu kod doğru olduğu sürece, döngü devam eder, eğer koşul sağlanmıyorsa döngüden çıkılır.

Üçüncü parçada yer alan kodlar ise; yine döngünün her turunda, süslü parantezin içindeki kodlar gibi çalışır. Tek fark, üçüncü parçadaki kodların başlangıcı, döngünün içinde yer alan kod parçalarının çalışmasından sonra olur.

Basit bir örnek vericek olursak;

```
[crayon-5c3f43b90cd06503258863/]
```

Ekran çıktısı aşağıdaki gibidir.

[crayon-5c3f43b90cd13156360858/]

For döngüsü ile ilgili bilinmesi gereken bir kaç küçük detay:

- For döngüsünü kullanırken eğer döngü içinde bir satır kod çalıştırılacaksa süslü parantez {} kullanılmasına gerek yoktur. Ancak birden fazla satır döngü içinde çalıştırılacaksa kesinlikle süslü parantez{} kullanılmalıdır.
- For döngüsünün parantez kısmında her bölüme bir şey yazmak zorunda değildir. Örneğin hepsi boşta bırakılabilir ya da başlangıç değeri kısmını boş bırakılıp döngünün dışında başlangıç değeri atayıp, ikinci ve üçüncü kısımları doldurulup döngü oluşturulabilir.
[crayon-5c3f43b90cd21084851771/]
- For döngüsü döngüyü oluşturan değişkenleri tek bir satırda toplar. Dolayısıyla döngü yapısını daha rahat okumak ve değişkenlere değer atanması ve güncellenmesi gibi komutların unutulmasında ortaya çıkabilecek hataların önüne geçebilmek açısından programcıya kolaylık sağlar.

For döngüsünü gerektiği zamanlarda iç içe de kullanabiliriz.

[crayon-5c3f43b90cd2d177315408/]

Ekran çıktısı aşağıdaki gibidir.

[crayon-5c3f43b90cd3a379288779/]

Örnek:

[crayon-5c3f43b90cd48611162090/]

Ekran çıktısı aşağıdaki gibidir.

[crayon-5c3f43b90cd56598319702/]

While Döngüsü

While çok kullanılan tekrarlama yapılarından biridir. Belirli bir mantıksal deyim sağlandığı sürece, belirli bir işin tekrarlanması isteniyorsa, bu denetim yapısı kullanılır. Bu yapıda, tekrarlamanın kaç kez olacağını önceden bilmemiz gerekmez. While döngüsünün yapısı aşağıdaki gibidir.

```
[crayon-5c3f43b90cd63113503970/]
```

While döngüsünde koşulun en başta olması ayırt edici bir özelliktir. Program akışı önce buradaki koşulu mantıksal bir süzgeçten geçirir. Buna göre eğer koşula yazılan ifade matematiksel olarak true (doğru) değer döndürüyorsa, süslü parantezlerle çevrili bloktaki komutlar sırasıyla çalıştırılır. Eğer programda boolean işlemi sonuclandıracak bir ifade yoksa bu döngü sonsuza kadar devam edebilir. Ancak false (yanlış) değerini döndürüyorsa, bu komutlar hiçbir zaman çalışmayacaktır. Bu durumda program akışına döngünün bittiği yerden devam eder. Burada while döngüsünün kilit noktası, döngünün hiç çalışmayabilir olmasıdır.

Örnek:

```
[crayon-5c3f43b90cd70581570081/]
```

Programın çıktısı aşağıdaki gibidir.

```
[crayon-5c3f43b90cd7e375480804/]
```

While döngüsünün çalışma mantığı : Program while döngüsünün bulunduğu satıra geldiği zaman önce ifade sonucunun doğru olup olmadığını kontrol eder. Eğer ifade yanlış bir sonuç verirse döngüye hiç girmeden bir sonraki satırdan çalışmasına devam eder. Eğer ifade doğru bir sonuç verirse, döngüde yer alan ifade yanlış bir sonuç verene kadar, döngü içinde yer alan işlem satırlarını çalıştırır. Yani, ifade doğru olduğu sürece döngü çalışmasına devam eder. while döngüsünde ifadenin sonucu döngünün başlangıcında kontrol edildiğinden, ifadenin yanlış sonuç verdiği durumlarda döngü bir defa bile çalışmaz.

While döngüsünde, for döngüsünden farklı olarak, ilk değer atama ve artırma/azaltma bölümleri bulunmamaktadır. While döngüsünde ilk değer atama işlemi döngü öncesinde, artırma/azaltma işlemi ise döngü kod bloğu içinde yapılır.

Do-While Döngüsü

Do-While döngü yapısı ile While döngü yapısı birbirine çok benzer. Önemli farklardan biri do-while yapısında döngü en az bir kere tekrarlanır . Bunun nedeni döngü kontrolünün döngünün sonunda yapılmasıdır. Ama while döngü yapısında kontrol en başta olduğu için, döngü hiç dönmeyebilir. Bu yüzden do-while yapısı kullanırken döngünün en az bir kere döneceği unutulmamalıdır. Diğer önemli fark ise while döngüsünde ilk önce şart sağlanıyor mu diye kontrol edilip ona göre işlemler gerçekleştirilirken. Do while döngüsünde ilk önce işlem gerçekleştirilip daha sonra şart sağlanıyor mu diye kontrol edilir.

Do-While döngüsünü bir kodun kesin bir kez çalışması gerektiği yerlerde kullanılabilir. Do-While döngüsünün yapısı aşağıdaki gibidir:

```
[crayon-5c3f43b90cd8c651927685/]
```

Boolean ifade döngünün sonunda görülmektedir, bu yüzden döngünün içindeki komutlar Boolean test edilmeden önce bir kere çalışır. Boolean ifade true ise, kontrol akışı geri atlama yapar ve döngü içindeki komutlar tekrar çalışır. Bu işlem, Boolean ifade false olana kadar tekrarlanır.

Program do döngüsüne geldiğinde hiç bir koşula bağlı olmadan direk olarak döngüye giriş yapar. Döngünün içinde yer alan işlem satırlarını çalıştırır. Eğer döngünün son satırında yer alan while deyimini ile ilgili ifade doğru sonuç vermez ise döngü sona erer. Do döngüsünde eğer sadece tek bir işlem satırı tekrarlanacaksa ({}) işaretlerine gerek yoktur. Döngü, while satırındaki ifade sonucu doğru olduğu sürece çalışmasına devam eder.

Örnek:

```
[crayon-5c3f43b90cd9a359150028/]
```

Programın çıktısı aşağıdaki gibidir:

```
[crayon-5c3f43b90cda7453328457/]
```

While döngüsü ne verilen ifade doğru değil ise kod while döngüsü içerisine hiç girmeden sonraki satır ile işlemlere devam eder, do while döngüsü nde ise ifadenin doğru olup olmamasına bakmaksızın döngü bir sefer çalışır.

For, While ve Do While Döngülerinin Birlikte İncelenmesi

Döngülerle ilgili olarak 3 temel kavram bulunmaktadır:

1. İlk değer atama
2. Koşul
3. Arttırma(Azaltma)

- For döngüsünde bütün kavramlar for satırında yer alan parantezler içinde tanımlanabileceği gibi, ilk değer atama bölümü döngü öncesinde ve artırma/azaltma bölümü döngü kod bloğu içinde tanımlanabilir.
- While döngüsünde ilk değer atama bölümü döngü öncesinde, koşul bölümü while satırında yer alan parantezler içinde, artırma/azaltma bölümü ise koşul sağlandığı takdirde çalışan kod bloğunun içinde tanımlanır.
- Do döngüsünde ise, ilk değer atama bölümü döngü öncesinde, koşul bölümü döngü sonundaki while satırında yer alan parantezler içinde, artırma/azaltma bölümü ise döngüye bağlı kod bloğu içinde tanımlanır.
- Do döngüsünün diğer döngülerden en büyük farkı, koşul kontrolü döngü sonunda yapıldığı için, koşul sağlanmasa bile döngünün en az bir kez çalışmasıdır. for ve while döngüleri ise, koşul sağlanmadığı takdirde hiç çalışmaz.

Kaynakça

1. <http://www.baskent.edu.tr/~tkaracay/etudio/ders/prg/java/ch09/loops.htm>
2. <https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-101/for-dongusu>
3. <http://www.yazilimutfagi.com/10373/dersler/java/javada-dongu-kullanimi-for-while-ve-do-while-donguleri.aspx>
4. <https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-101/for-dongusu>
5. <http://immibilisim.com/blog/236/javada-kontrol-yapilari-ve-dongu-kaliplari>
6. <https://www.mobilhanem.com/temel-java-dersleri-while-dongusu/>
7. <https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-101/while-dongusu>
8. http://www.bilgigunlugum.net/java/2java_dongu.html
9. <https://emraharslanbm.wordpress.com/tag/do-while-dongusu/>
10. http://javayaz.com/?page_id=75