

# PART 14

## 14. Düzenli İfadeler

Java, düzenli ifadeler ile kalıp eşleşmesi için `java.util.regex` paketini sağlar. Java düzenli ifadeleri, Perl programlama dili ile çok benzerdir ve öğrenmesi kolaydır.

Bir düzenli ifade, desen içinde tutulan özelleştirilmiş bir syntax kullanılarak, diğer stringleri ve string kümelerini eşleştirmeye veya bulmanıza yarayan bir özel karakter dizisidir. Metin ve veriyi aramak, düzenlemek veya değiştirmek için kullanılırlar.

`Java.util.regex` paketi başlıca ,aşağıdaki üç sınıftan oluşur.

- **Pattern Sınıfı:** Bir pattern nesnesi, düzenli ifadenin derlenmiş gösterimidir. Pattern sınıfı, public constructor sağlamaz. Kalıp oluşturmak için, öncelikle onun public static derleme metotlarını çağırmanız gerekmektedir, bu bir Pattern nesnesi döndürecektir. Bu metotlar , düzenli ifadeyi ilk bağımsız değişken olarak kabul eder.
- **Matcher Sınıfı:** Bir eşleştirici nesnesi, deseni yorumlayan ve bir input string'ine karşı eşleştirme işlemlerini gerçekleştiren bir motordur. Pattern sınıfı gibi, Matcher sınıfı da public constructor tanımlamaz. Bir Pattern nesnesi üzerinden matcher metodunu çağırarak bir Matcher nesnesi elde edebilirsiniz.
- **PatternSyntaxException:** Bir PatternSyntaxException nesnesi, düzenli ifade desenindeki syntax hatasını gösteren denetimsiz bir exception'dır.

### Örnek

Aşağıdaki örnek verilen stringten e-mail stringlerinin nasıl bulunacağını anlatmaktadır:

```
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class RegexMatches
{
    public static void main( String args[] ){

        // String to be scanned to find the pattern.
        String line = "Benim mailim maltepe@maltepe.edu.tr Bu ise arkadaşımın maili
aa@aa.com şeklindedir.";
        String pattern = "\\b[\\w.%-]+@[-.\\w]+\\. [A-Za-z]{2,4}\\b";
```

```
// Create a Pattern object
Pattern r = Pattern.compile(pattern);

// Now create matcher object.
Matcher m = r.matcher(line);
while (m.find( )) {
    System.out.println("Found:" + m.group() );
}
}
```

Bu aşağıdaki sonucu üretecektir:

```
Found: maltepe@maltepe.edu.tr
Found: aa@aa.com
```

Group index değerini kullanmak için

```
// Now create matcher object.
Matcher m = r.matcher(line);
m.find();

System.out.println("Found 0:" + m.group(0) );
```

Yukarıda verilmiş kod parçası, sadece ilk bulunmuş string'i gösterecektir.

## 14.1 Düzenli İfadelerin yazım biçimi

Buradaki tablo, Java dilinde mevcut bütün düzenli ifade metakarakter syntax'ını listelemektedir.

Subexpression	Matches
^	Satır başlangıcını eşleştirir.
\$	Satır sonunu eşleştirir.
.	Yeni satır hariç herhangi bir tekil karakterini eşleştirir. M seçeneğini kullanmakta yeni satırı eşleştirmeyi sağlar.
[...]	Köşeli parantez içinde herhangi bir tekil karakterini eşleştirir.
[^...]	Köşeli parantez olmayan herhangi bir tekil karakterini eşleştirir.
\A	Bütün string'in başlangıcı
\Z	Bütün string'in sonu
\Z	İzin verilen son satır sonlandırıcı hariç bütün string'in sonu
re*	Önceki ifadenin 0 veya daha fazla olayını eşleştirilir.

re+	Önceki şeyin 1 veya daha fazlasını eşleştirir.
re?	Önceki ifadenin 0 veya 1 olayını eşleştirir.
re{ n}	Önceki ifadenin n sayıda olayını eşleştirir.
re{ n,}	Önceki ifadenin n veya daha fazla olayını eşleştirir.
re{ n, m}	Önceki ifadenin en az n ve en fazla m olayını eşleştirir.
a  b	A veya B den birini eşleştirir.
(re)	Düzenli ifadeleri gruplar ve eşlenmiş metni hatırlar.
(?: re)	Düzenli ifadeyi, eşlenmiş metni hatırlamadan gruplar.
(?> re)	Bağımsız deseni geileme yapmadan eşleştirir.
\w	Kelime karakterlerini eşleştirir.
\W	Kelime olmayan karakterleri eşleştirir.
\s	Alfabe dışı karakteri eşleştirir. [\t\n\r\f] eşdeğer.
\S	Matches nonwhitespace. Alfabe dışı olmayan karakteri eşleştirir..
\d	Basamak eşleştirir. [0-9] eşdeğer
\D	Sayı olmayanı eşleştirir.
\A	String başlangıcını eşleştirir.
\Z	String sonunu eşleştirir. Eğer yeni satır varsa, yeni satırdan hemen öncekini eşleştirir.
\z	String sonunu eşleştirir.
\G	Son eşleşmenin bittiği noktayı eşler
\n	Grup numarası "n" olan back to reference yakalar
\b	Köşeli parantezin dışındaki kelime sınırlarını eşleştirir. Köşeli parantezin içindeki geri (backspace (0x08)) karakterlerini eşleştirir.
\B	Kelime olmayan sınırları eşler.
\n, \t, etc.	Yeni satırları, carriage returnleri, tablaları eşler
\Q	\E ye kadar olan kaçış karakterleri
\E	\Q ile başlamış alıntıyı bitirir.