

# JAVA 14 New Features

JDK13 ten sonra nerdeyse hiçbir yenilik yok 😊 Kabuller var. O yüzden henüz indirmedim bilgisayarıma.

- Pattern Matching for instanceof (Preview)

Typecast işlemleri yaparken önceden şöyle yapardık(jdk14-). Şimdi ve sonrasında böyle yapacağız(jdk14+). Çok fark yok ama olsun 6 ayda bir sürüm yeni jdk sürümü çıkınca böyle değişiklikler normal 😊

Exp: jdk14-

Exp: jdk14+

```
if (obj instanceof String) {
    String s = (String) obj;
    // use s in your code from here
}
```

```
if (obj instanceof String s) {
    // can use s directly here
}
```

- Switch Expression Is Ok

\* Oracle Corp. Java dünyasına preview olarak getirdiği özellikleri Developer' lar tarafından benimsenecek mi diye feedback bekleyerek getiriyor. Sonraki sürümlerde iyileştirmeler veya vazgeçmeler olabilir. Jdk12 de başlayan **switch expression** preview 13 de' de devam ederek **yield** yapısı eklenmişti. **Jdk14 te Kabul görmüş durumda.**

```
var log = switch (event) {
    case PLAY -> "User has triggered the play button";
    case STOP, PAUSE -> "User needs a break";
    default -> {
        String message = event.toString();
        LocalDateTime now = LocalDateTime.now();
        yield "Unknown event " + message +
            " logged on " + now;
    }
};
```

- Text Blocks New Features

Jdk13 te gelen multiline text yapısına bir önceki döküman da değinmiştik. Çok rahat şekilde html, json vb. ifadeleri text olarak satır satır yazabiliyoruz.

Jdk14 te textblock yapısına iki özellik eklenmiş. '**\**' ve '**\s**' karakteri ile kullanımlar gelmiş.

Şöyle ki:

```
String literal =
    "Lorem ipsum dolor sit amet, consectetur adipiscing " +
    "elit, sed do eiusmod tempor incididunt ut labore " +
    "et dolore magna aliqua.";
```

gibi eski jdk larda yazdığımız bir text' i

```
String text = ""  
    Lorem ipsum dolor sit amet, consectetur adipiscing \\  
    elit, sed do eiusmod tempor incididunt ut labore \\  
    et dolore magna aliqua.\\  
    "";
```

şeklinde yazabiliyoruz. \ karakteri satır sonu vererek sonraki satıra geçmemizi sağlıyor yani.

\s karakteri ile Unicode bazında bir boşluk veriyoruz. ☺ harika özellik tam çıldırmalık.

- **Helpful NullPointerExceptions**

Bu güzel olmuş.

Developer' ların en sevmediği ve görmek istemediği throw exception bence nullpointer :D acımasız hain bir hata. Hatta vatan haini :D

```
var name = user.getLocation().getCity().getName();
```

Mesela böyle bir kullanımda. User in location, city üzerinden de name bilgisini edinmek istediğimizde. Null alıyorsak. Sebebi neydiki. Diye düşündüğümüzde. Name null mı city mi null diye debug yaparız. Ama throw exception da bunun detayı verilse. Belki user in kendisi null. Hiçbir şey get edilemiyor belki. Yani şöyle ki

```
Exception in thread "main" java.lang.NullPointerException  
    at NullPointerExceptionExample.main(NullPointerExceptionExample.java:5)
```

böyle bir hata mesajındansa:

```
Exception in thread "main" java.lang.NullPointerException: Cannot invoke  
"Location.getCity()" because the return value of "User.getLocation()" is null
```

Bu mesaj daha hoş.

Bu özelliği aktif yapmak için

```
-XX:+ShowCodeDetailsInExceptionMessages
```

Flag ini properties e eklemek veya vm options üzerinden param geçmek gerekmekte.

**Melih Hilmi ULUDAĞ**



Kaynaklar

<https://openjdk.java.net/projects/jdk/14/>

<https://blogs.oracle.com/javamagazine/java-14-arrives-with-a-host-of-new-features>