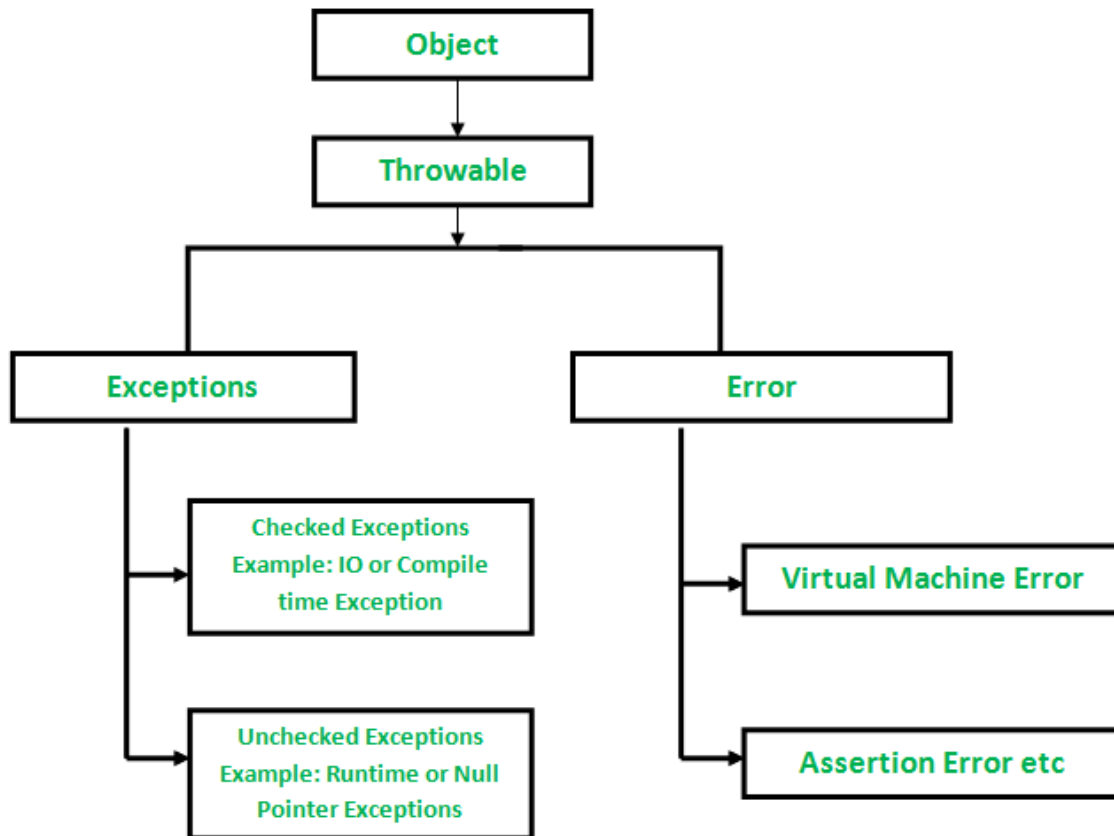


Checked ve UnChecked Exception



Unchecked Exception

RuntimeException sınıfından kalıtılmış exceptionlardır. Derleme zamanında bizi herhangi bir exception handling için zorlamaz yani derleme zamanında kontrol edilemeyen exceptionlardır. RuntimeException sınıfından bir exception fırlatıldığında onu try catch içine almazsam herhangi bir uyarı vermez. Derleme zamanında kontrol edilmediği için kod başarılı derlense bile çalışma zamanında hatalarla karşılaşırız.

Exception sınıfından unchecked ve checked sınıfındaki exceptionlar kalıtılmıştır. Burda unchecked olan hataları try catch bloklarını özelleştirerek yakalayabilmemizi sağlar.

Unchecked exceptionlara örnek olarak NullPointerException, ArrayIndexOutOfBoundsException ve IllegalArgumentException verebiliriz.

```
String email = null;
String sender = "ozturkk.halimee@gmail.com";

if(email.equals(sender)) {
    System.out.println("Equals");
}
```

Yukarıdaki kod blogu NullPointerException atar.

```
Exception in thread "main" java.lang.NullPointerException Create breakpoint  
at Main.main(Main.java:13)
```

Bu exceptionı try catch içine alarak da yakalayabiliriz. Bu unchecked için çok önerilen bir yöntem değil.

```
String email = null;  
String sender = "ozturkk.halimee@gmail.com";  
try{  
    if(email.equals(sender)){  
        System.out.println("Equals");  
    }  
}catch (NullPointerException e){  
    System.out.println("NullPointerException" + e);  
}
```

Kodu bu şekilde yazınca NullPointerException hatasını istediğimiz şekilde handle edebiliriz.

NullPointerException almamak için kodu fixlemeye çalışalım. Aşağıdaki şekilde NullPointerExceptionın önüne geçmiş oluruz.

```
String email = null;  
String sender = "ozturkk.halimee@gmail.com";  
if(email != null && email.equals(sender)){  
    System.out.println("Equals");  
}
```

Checked Exception

Derleyici tarafından kontrol edilen exception türüdür. Derleme zamanında yakalanıp handle edilmesi gerekir. Derleyici tarafında kontrol edildiği için biz bu exceptionları kodumuzda belirtmeliyiz. Try catch blogunda hataları yakalamamız gerekir. Kod hataları handle edene kadar derlenmeyecektir.

Örneğin

```
FileReader file = new FileReader("example.txt");
```

Bu kodu yazdığımda ve çalıştırmaya çalıştığımda kod derlenmiyor ve hata veriyor.

```
java: unreported exception java.io.FileNotFoundException; must be caught or declared to be thrown
```

Hata da açık bir şekilde handle etmemiz gerektiğini söylüyor.

```
try {  
    final FileReader file = new FileReader("example.txt");  
} catch (final FileNotFoundException e) {  
    e.printStackTrace();  
}
```

Kodu bu şekilde handle ettiğimizde kodumuz başarıyla derlenecektir. Başka bir yolda şu şekildedir.

```
public void throwsExample() throws IOException {  
    FileWriter fw = new FileWriter("file.txt");  
}
```

Checked Exceptionlara örnek olarak IOException, SQLException, ClassNotFoundException verilebilir.

Kaynakça:

<https://www.geeksforgeeks.org/checked-vs-unchecked-exceptions-in-java/>

<https://www.okanuzun.com/java-checked-ve-unchecked-exception/#:~:text=%C4%B0sminden%20de%20anla%C5%9F%C4%B1laca%C4%9F%C4%B1%20gibi%20derleyici,Exceptionlar%20RuntimeException'%C4%B1n%20alt%20s%C4%B1n%C4%B1flar%C4%B1d%C4%B1r.>

<https://figencan.wordpress.com/2014/07/14/javada-exception-handling-kavrami/>

<https://hkdemircan.medium.com/checked-and-unchecked-exception-5686ba7a6721>

<https://rollbar.com/blog/how-to-handle-checked-unchecked-exceptions-in-java/#:~:text=Difference%20Between%20Checked%20and%20Unchecked%20Exceptions%20in%20Java,-To%20summarize%2C%20the&text=unchecked%20exception%20is%3A-,A%20checked%20exception%20is%20caught%20at%20compile%20time%20whereas%20a,t%20required%20to%20be%20handled.>