

Exception

- Runtime sırasında oluşan hatalara verilen isim.
 - Program çalışırken oluşan.
- Diğerleri compile-time hatalar.
- Exception handling
- Try Catch
- Throw

Try catch

```
{
    //hesaplanmak istenen ifade
}
catch
{
    //Bir hata türü tespit edilince verilmesi gereken mesaj
}
catch
{
    //başka Bir hata türü tespit edilince verilmesi gereken mesaj
}
finally
{
    //her durumda çalıştırılacak olan kod parçası
}
```

Try içinde çalıştırmak istediğimiz şeyi yazarız. Eğer bu çalışırsa, catch'leri atlar ve **finally**'e geçer.

Eğer try içindeki çalışmıyorsa, o zaman **catch**'e geçer. Ardından **finally** çalışır.

Try catch

Finally bloğu her durumda çalışır. Ancak onu yazmak zorunda değiliz.

- Aynı zamanda iki tane *catch* koymak zorunda da değiliz.

```
//main
try {
    int a = 5/0;
} catch (ArithmeticException e){
    print("hata oluřtu", e.getMessage());
}
```

Kullanılacak birçok Exception türü bulunur. Duruma göre onlardan birini seçebilirsiniz.

```
catch (Exception e)
```

Örneğin bir int array oluşturup, olmayan elemana ulaşmak isteyelim.

Try catch

```
public static void main(String[] args) {  
  
    try {  
        int[] x = new int[3];  
        System.out.println(x[0]);  
  
    }  
    catch(Exception e) {  
        System.out.println("error here" + e.getMessage());  
  
    }  
    finally {  
        System.out.println("her türlü");  
    }  
  
}
```

Throw (Hata Fırlatma)

Bazı durumlarda programı bir şekilde kontrol altına almak gerekiyor. Try catch bunlardan biri. Onu kullanırken biz **Exception** e yazarak atabildik.

- Ancak eğer bunu *throw* ile kullanmak istersek o zaman işler değişiyor.
- Throw ile atabileceğimiz hazır Exceptionlar var, onları atarsak sorun yok.
 - Ancak eğer hazır Exceptionlardan birini kullanmayacaksak o zaman bunu tepeye **throws Exception** olarak yazmak gerekiyor.

Java dosya operasyonları

Dosyaya yazmak ve okumak için `BufferedReader` kullanacağız. Bununla birlikte `FileWriter` da kullanılacak.

- Bunlar Java ile birlikte gelirler.
- Bu özellikleri kullanmak için try/catch kullanırız.
- Ya da tepede **throws** keyword'ünü de kullanabiliriz.

```
String str = "Bunu dosyaya yazdir";

File file = new File("dosya.txt");
if (!file.exists()) {
    file.createNewFile();
}

FileWriter fileWriter = new FileWriter(file, false);
BufferedWriter bWriter = new BufferedWriter(fileWriter);
bWriter.write(str);
bWriter.close();
```

Dosya

- **false** dediğimiz *append* ile ilgili.
- Bu kodu bu şekilde çalıştıramayız. Ya bazı şeyleri try catch içine alacağız ya da tepede olası bir hatayı throw edebilmek için bundan bahsedeceğiz.
- Dosya okumak için aşağıdaki kodu kullanıyoruz:

```
FileReader fileReader = new FileReader(file);
String line;

BufferedReader br = new BufferedReader(fileReader);

while ((line = br.readLine()) != null) {

    System.out.println(line);

}

br.close();
```

Dosya kopyalama

Zaten dosya okuduk. Şimdi üstteki koda Writer ekleyeceğiz, ekrana yazdırana kadar o dosyaya yazacağız `write()` ile.

Referanslar

- <https://gelecegiyazanlar.turkcell.com.tr/konu/egitim/android-101/trycatchfinally-yapisi>
-