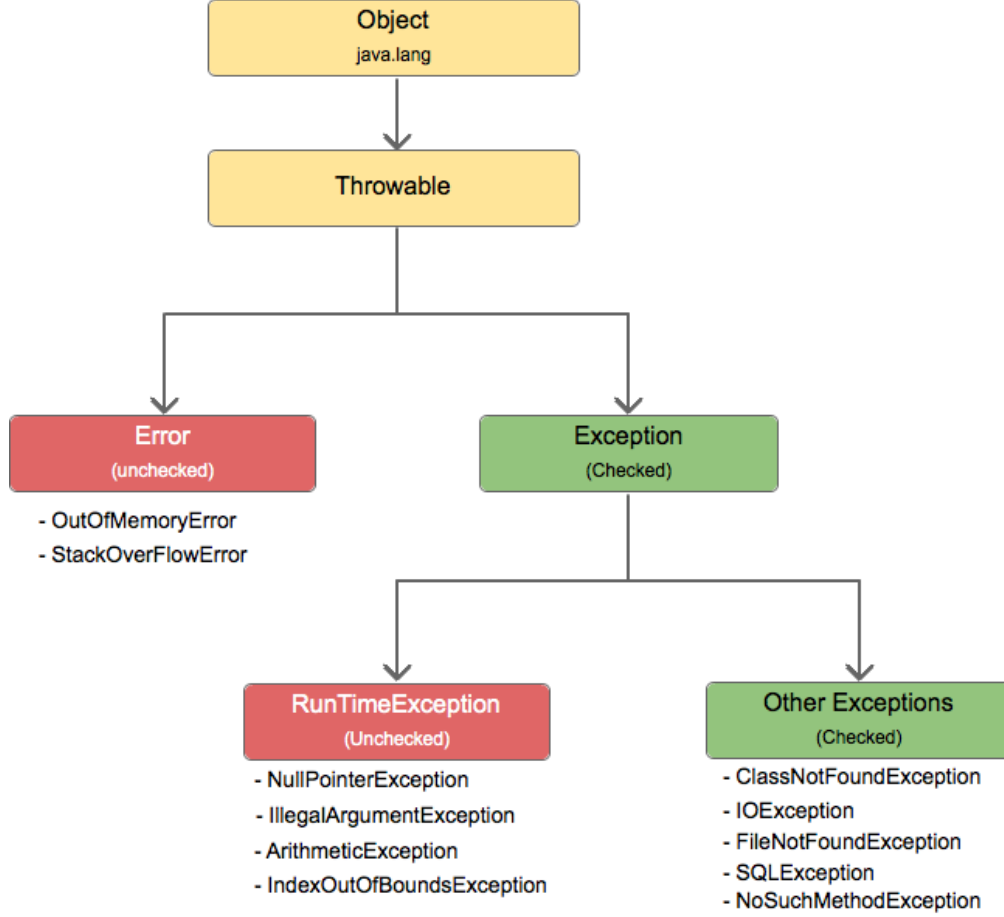


UNCHECKED EXCEPTION ve CHECKED EXCEPTION NEDİR ?

Java dilindeki exception haritasına bakarak yazımıza başlayalım ve daha sonra iki kavramı inceleyelim.



UNCHECKED EXCEPTION

Adından da anlaşılacağı üzere bu exceptionlar derleyici tarafından yakalanamazlar. Kodumuz başarıyla derlendikten ve çalışmaya başladıktan sonra ilgili satırda exception fırlatır.

Bilinen en iyi unchecked exception örneği **NullPointerException**dur.

Kodumuz çalışır fakat ilgili nesne yaratılmadıysa yani hala null ise kod bu noktada hata fırlatır.

ÖR:

```

1
2
3 public AdvertResponse convertToAdvertResponse(Advert savedAdvert) {
4     AdvertResponse response;
5     response.setBaslik(savedAdvert.getBaslik());
6     response.setFiyat(savedAdvert.getFiyat());
7     response.setAdvertNo(savedAdvert.getAdvertNo());
8     return response;
9 }
10
11

```

Örnekte parametre olarak gelen Entityimizi response nesnesine dönüştürmeyi amaçlıyoruz.

Fakat response nesnesi yaratılmadığı yani new anahtarlı keywordu ile başlatılmadığı için burada NullPointerException hatası alacağız.

CHECKED EXCEPTION

Bu tür exceptionlar derleyici tarafından kontrol edilen exceptionlardır. Derleyici bu hataları fark ettiğinden eğer gerekli noktalarda exceptionlarımızı fırlatmadıysak derleme işlemi tamamlanmaz ve kodumuzu çalıştıramayız.

Checked exceptionlarımızı try catch bloğu içerisinde handle etmemiz gerekmektedir.

Bilinen en yaygın checked ben exceptionlar IOException, SQLException olarak söylenebilir.

ÖR:

```

1
2 protected Connection connect() {
3
4     try {
5
6         Class.forName(JDBC_DRIVER);
7         return DriverManager.getConnection(DB_URL, USER, PASS);
8
9     } catch (SQLException | ClassNotFoundException e) {
10        // TODO Auto-generated catch block
11        e.printStackTrace();
12    }
13
14    return null;
15
16 }
17

```

Örnekte JDBC driver aracılığıyla bir db connect methodu yazılmıştır. Fakat burada bağlantı sırasında bir hata ile karşılaşırsa Exception fırlatması üzerine bir yapı kurulmuştur.