

EXCEPTIONS

İki tür exceptions vardır.

1- Compile Time (checked) Exceptions

Kod yazıldığında Java'nın öngördüğü olası sorunlardır. Olası bir problemde altını kırmızı çizer, fakat her kırmızı çizgi exceptions değildir.

2-Run Time (unchecked) Exceptions

Kod çalıştırıldığında ortaya çıkan exceptions'lardır.

Compile Time Exceptions Types (checked)

1- FileNotFoundExceptions:

Javaya bir dosyayı okumasını veya yazmasını istediğimizde eğer bulamam diye korkarsa (dosya silinmiş yada yanlış yazılma ihtimali) FileNotFoundExceptions verir. Bu hatayı try-catch ile (ya da class isminin yanına yazdığımız throws ile –ki bu çok tercih edilmez) çözmedikçe Java CTE verir.

2- InputOutputExceptions (IOExceptions):

Dosyayı bulmasına rağmen dosyayı formatından dolayı okuyamıyorsa, IOExceptions verir. Genel yazma ve okuma sorunlarını handle eder.

Java.lang.Object	PARENT	
Java.lang.Throwable	^	
Java.lang.Exceptions	^	
Java.io.IOExceptions	CHILD	ilişkisi
Java.io.FileNotFoundExceptions		

NOT: Nested try-catch bloğu içinde eğer bir parent-child ilişkisi varsa;

İlk child bloğu içine child class ı yazmalıyız. Eğer önce parent yazılır ve ikinci catch bloğuna child yazılırsa, child a iş kalmayacağından Java –erişilemez catch- bloğu diyerek CTE verir.

NOT: Parent-child ilişkisi varsa bu durumda child exceptions için catch bloğu yazmasakta olur, fakat bir sorun ile karşılaşırsa bunun dosya bulunamadığı için mi yoksa bulupta okuyamadığı için mi olduğunu anlayamayız.

Run Time Exceptions (unchecked):

1- ArithmeticExceptions:

Casting, conversion veya aritmetikte oluşan hatalardır. Bir sayının sıfıra bölünememesi yada sonsuz negatif sayı olmaması gibi.

2- NullPointerExceptions:

Null, var olmayan bir değeri ifade ettiği için null ile String kuralları ile bir işlem yapıldığında (String str=null; str.length yapıldığında), bu hata ile karşılaşırız. (String str2="can" verildiğinde, str+str1=nullcan olur, yani concatenation yapılır.)

3-IndexOutOfBoundsExceptions:

List ve Array de var olmayan bir index ile işlem yapılmak istenirse bu hata ile karşılaşırız. (int arr[]={1,2,3} için indexOf(3) sorulursa)

4-NumberFormatExceptions:

Eğer veri yanlış bir formatta ise konsolda bu hatayı verir. (String str= "123a45"; için syso(Integer.parseInt(str)) yani kelimeyi sayıya çevir dersek "a" yüzünden hata verir.)

5-ClassCastException:

Data casting yapıldığında büyük geniş bir datayı , daha küçük bir dataya çevirmek istersek bu hatayı verir. (Object num =40; için String numStr=(String)num; yapılamaz). Object ten String e çevrilemez.

6-IllegalArgumentException:

Eğer yazılan koddaki kurallara aykırı bir input olursa bu hata ile karşılaşılır. (bir yaşı -20 girilmesi gibi, Java gündelik olayları bilmez)

NOT= Try-catch ile Add throws declaration arasındaki fark (Interview sorusu)

--Add throws declaration yapıldığında kod bloke olur. Handle etme imkanı yoktur.

--Try-catch ile ise exceptions handle edilir kod bloke olmaz.

Warning: eğer method signature dan sonra birden fazla exception throw yapılacak ise yazılacak exceptionsların sıralaması önemlidir. Parent-child ilişkisine göre küçükten büyüğe doğru gidilir ve oraya virgöl bırakılır. Sadece parent exceptions yazsakta olur.

Throw ve Throws

Throw;

- 1- Belirgin bir şekilde exceptions fırlatmak için kullanılır.
- 2- Sadece bir exceptions için kullanılır.
- 3- Method içinde kullanılır.
- 4- Syntax olarak throw kelimesinden sonra instance variable

Throws;

- 1- Throws kelimesi bir veya birden fazla exceptions için araya virgöl konularak kullanılır.
- 2- Çoklu exceptions throw kullanılarak fırlatılabilir.
- 3- Throws keyword ü method signature sonrasında aynı satırda kullanılır.
- 4- Syntax olarak throws kelimesinden sonra exceptions class isimleri yazılır.

NOT: Try bloğu, tek başına çalışmaz. Ya catch bloğu yada **"finally"** bloğu ile çalışır.

Finally block her durumda çalışır. Genelde cloud veya database ile bir bağlantı varsa onu kesmek için kullanılır. Mesela exception oldu ve application durdu ama hala database ile bağlantı açık kalır. Ama aslında sisteme saldırı olmaması için (maddi külfet artmasın diye) kapatılmalıdır. Hem öngörülen hem de öngörülmeyen exceptions durumlarında da çalışır.

