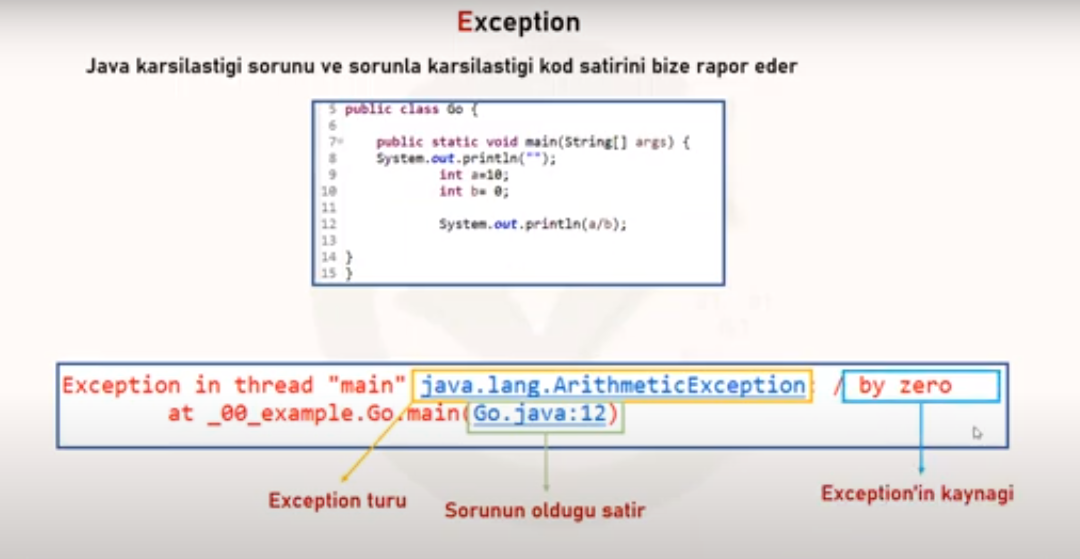
**EXCEPTION**

**(40.DERS)**

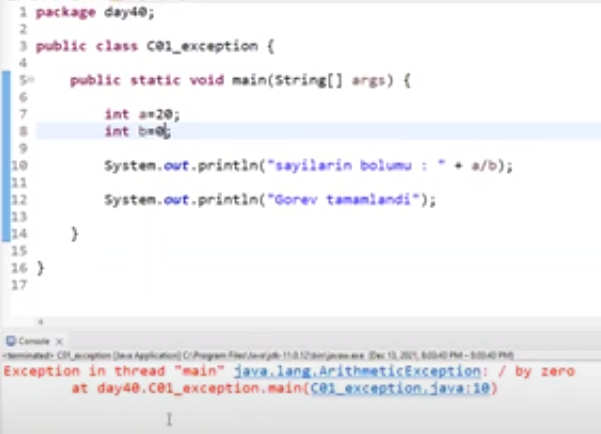
**(Java’nın Sorunlarla Mücadele Birimi)**

**Bir program çalıştırıldığında Java çözemediği bir sorunla karşılaştığında çalışmayı durdurur(stop exception) ve throws an exception**

**Java karşılaştığı sorunu ve sorunla karşılaştığı kod satırını bize rapor eder**

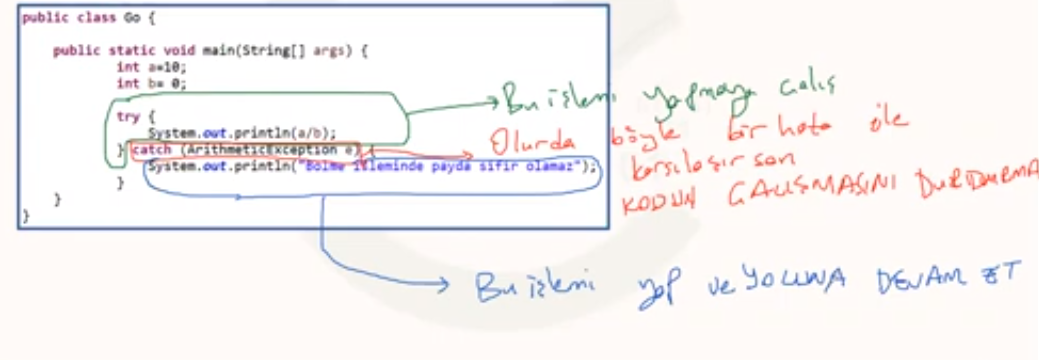


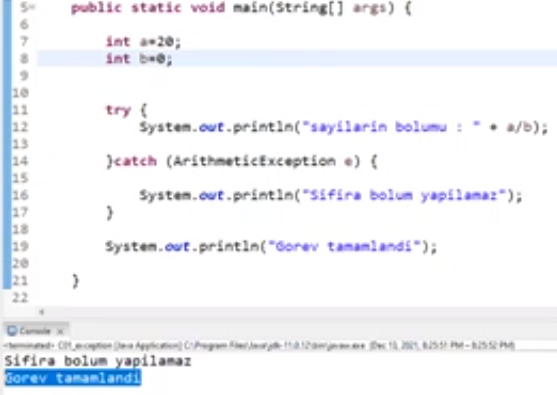
**Mesela sayı /sıfır matematik işlemde tanımsız olduğundan java kod çalışmayı durdurur yani exception la karşılaşır ve görevi bitirir. Bu yüzden syso yu da yazdıramaz yani biz javaya böyle bir sorunla karşılaştığında ne yapması gerektiğini söylememişiz.**



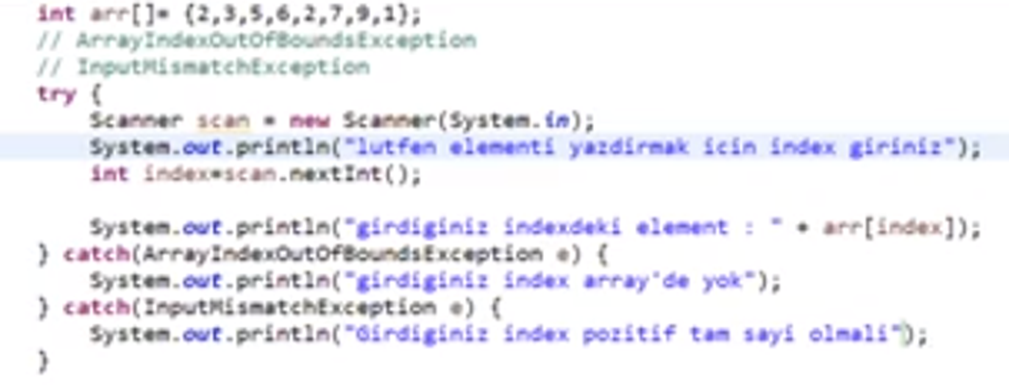
**Java karşılaştığı sorunu ve sorunla karşılaştığı kod satırını bize rapor eder**

**Sorunu çözmek için try-catch blok kullanırız. Ama try bloğu tek başına çalışmaz. Try bloğundan sonra catch blok(ları) veya finally blok olmalıdır.**



****

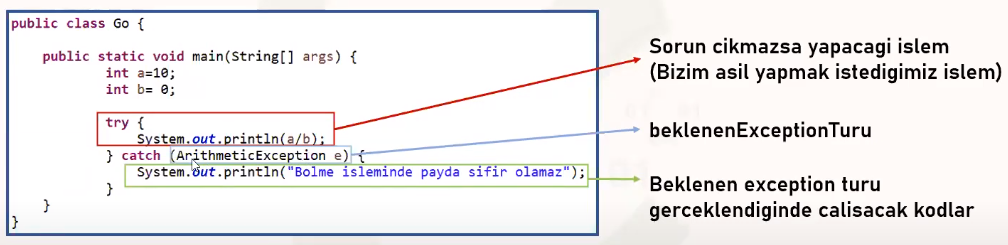
**Eğer birden fazla hatayla karşılaşırsak consolda görünen hatayı catch içine alıp yazdırmamız gerekir**

****

**Catch blokta kullanılan “e” nin görevi:**

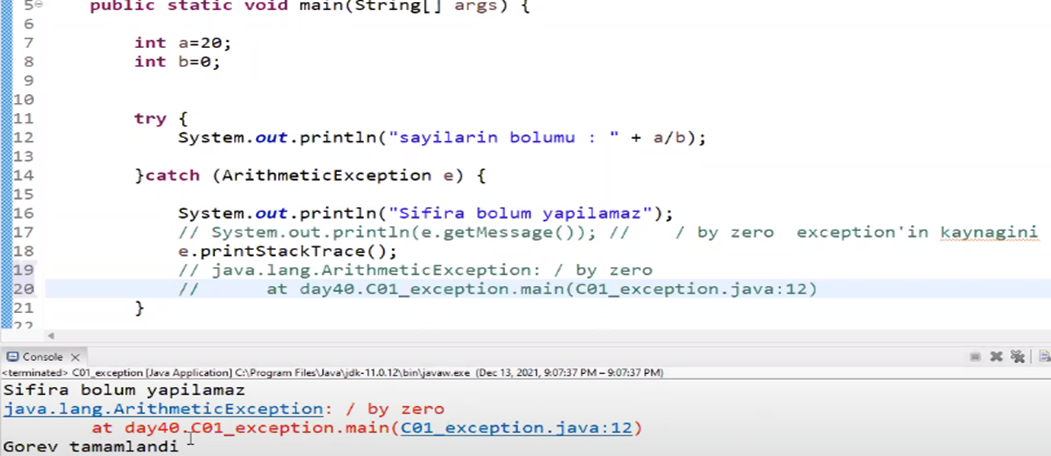
**Cath blokda yazdığımız Exception isimli class adı (data türü) , “e” ise variable ismidir.**

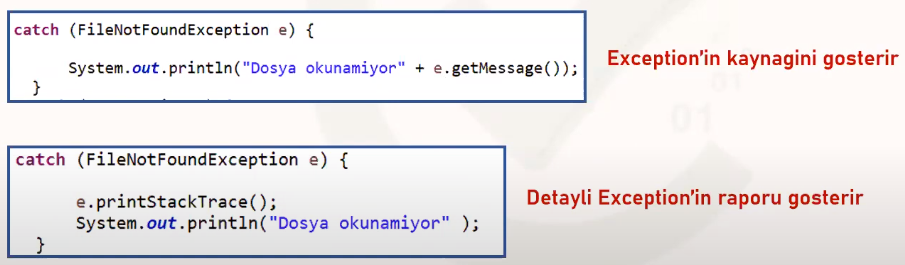
**Buradaki e ArithmeticException classından bir objedir.**

****

**e. yazınca ilgili exception classından kullanacağımız methodları görebiliriz ayrıca e. İle java bize ne yazdırıyor onu görebiliyoruz; e yazmadığımızda javaya biz neyi istiyorsak onu yazdırıyorduk.**

**Javanın bildirdiği hatayı da görelim istiyorsak e.getMessage yazarız. Böylelikle hem hatanın nereden kaynaklandığını görmüş oluruz hem de java kodu yazdırmaya devam etmiş olur.**

****

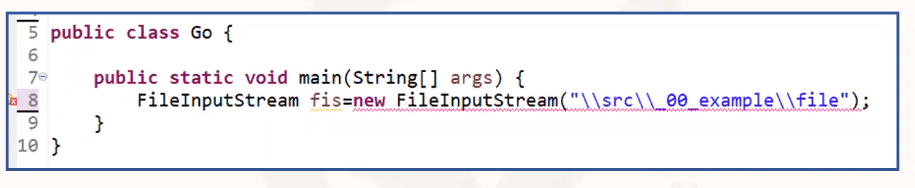
****

**Exception Türleri:**

**File Input/ Output Exception**

**Javada bir dosyaya ulaşmak istediğimizde file Input Stream den yardım alırız.**

**Java bu kodla olası bir problem öngörüyor. Ve bu probleme karşı ne yapması gerektiğini bize soruyor.**

****

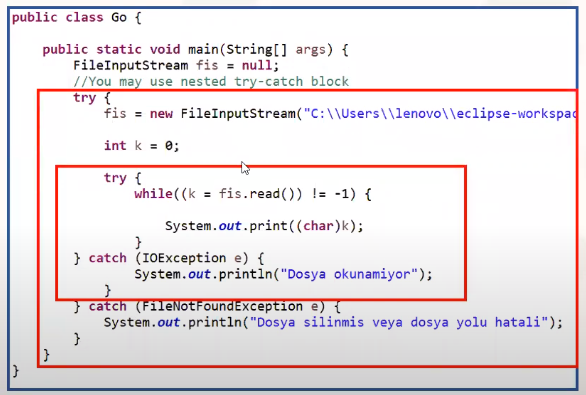
**Buradaki CTE kodumuzda bir hata olduğu için değil, yazdığımız kod çalıştığında oluşabilecek olası okuma hatalarında ne yapılacağına karar vermek içindir. Yani altı çizili her kod CTE değildir.**

**Bunun için iki durum söz konusu:**

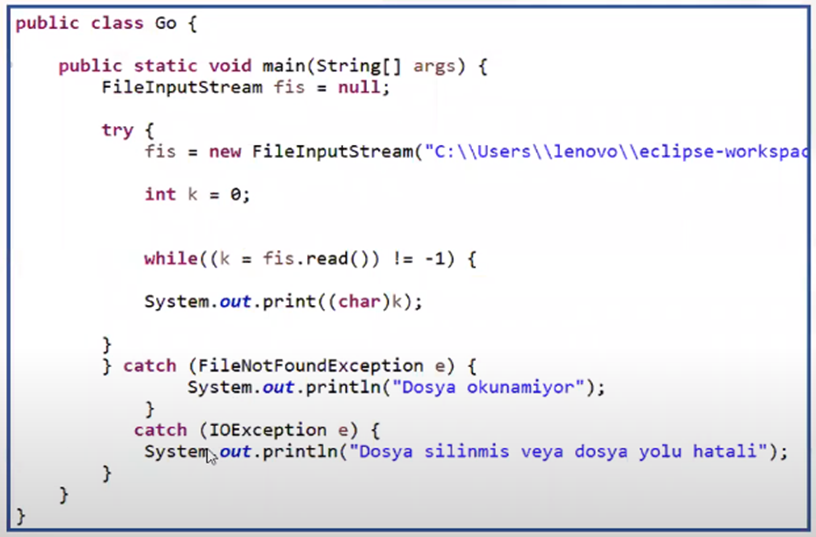
1. **Try catch kullanmak bu problemi Handle edip javanın yoluna devam etmesini sağlayabiliriz**
2. **Eğer dosya okunamıyorsa kod çalışmasın istiyorsak olayın farkında olduğumuzu ve sorumluluğun bizde olduğunu javaya söylemeliyiz**

**Muhtemel sorunlar birden fazla ise;**

1. **İçiçe try-catch blokları kullanılabilir**

****

1. **Tek try -multiple catch kullanılabilir.**

****

**Birden fazla catch blok kullanılacaksa yazılacak exceptionların sırası önemlidir. Birbiri ile parent child ilişkisi olan exceptionlar ise önce child olan yazılmalıdır. Aksi durumda child exception kullanılmaz olur.**

**Null Pointer Exception**

**Null objesini uygun olmayan bir komutla kullanırsanız java null pointer exception verir.**

**ArrayindexOutBoundsException**

**Array veya List de olmayan bir index için işlem yapmak isterseniz java ArrayindexOutBounds Exception verir.**

**Class CastException**

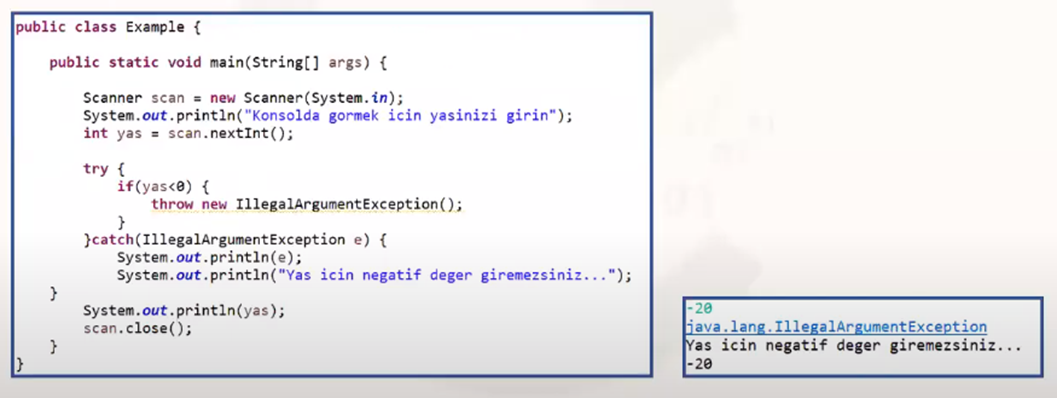
**Bir datayı casting yapılamayacak bir dataya çevirmek istediğinizde Class CastException verir**

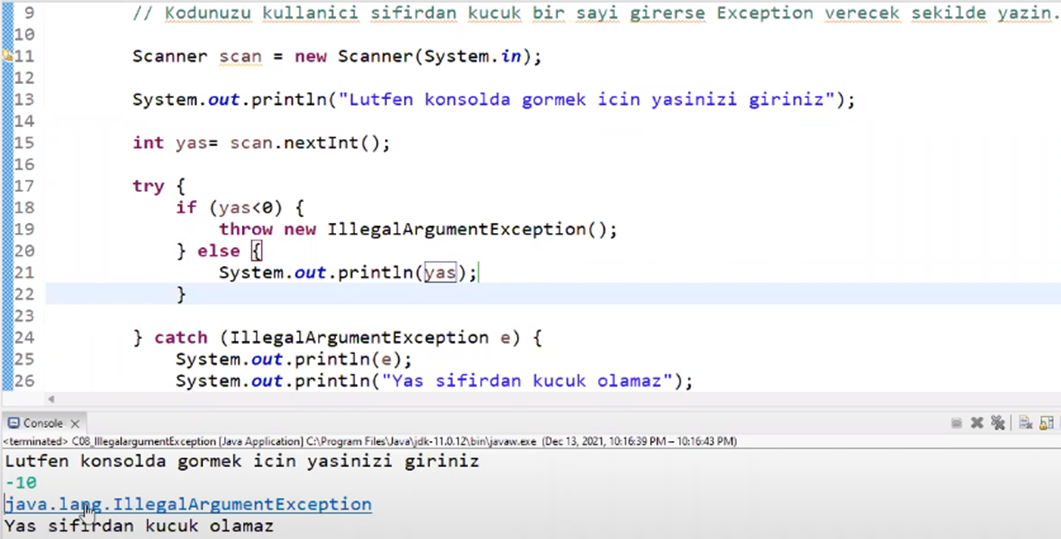
**NumberFormatException**

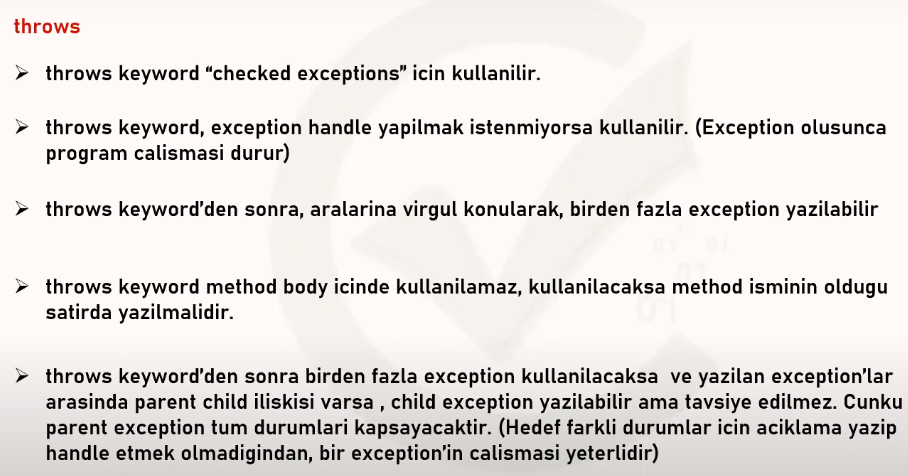
**sayı olmayan bir Stringi sayıya çevirmeye çalışırsanız java NumberFormatException verir**

**İllegal Argument Exception**

**Diğerlerinden farklı olarak bizim oluşturabileceğimiz bir exceptiondur.**

****

****

****