

Exception Handling

C++, Java, C# var.

throw → fırlatır

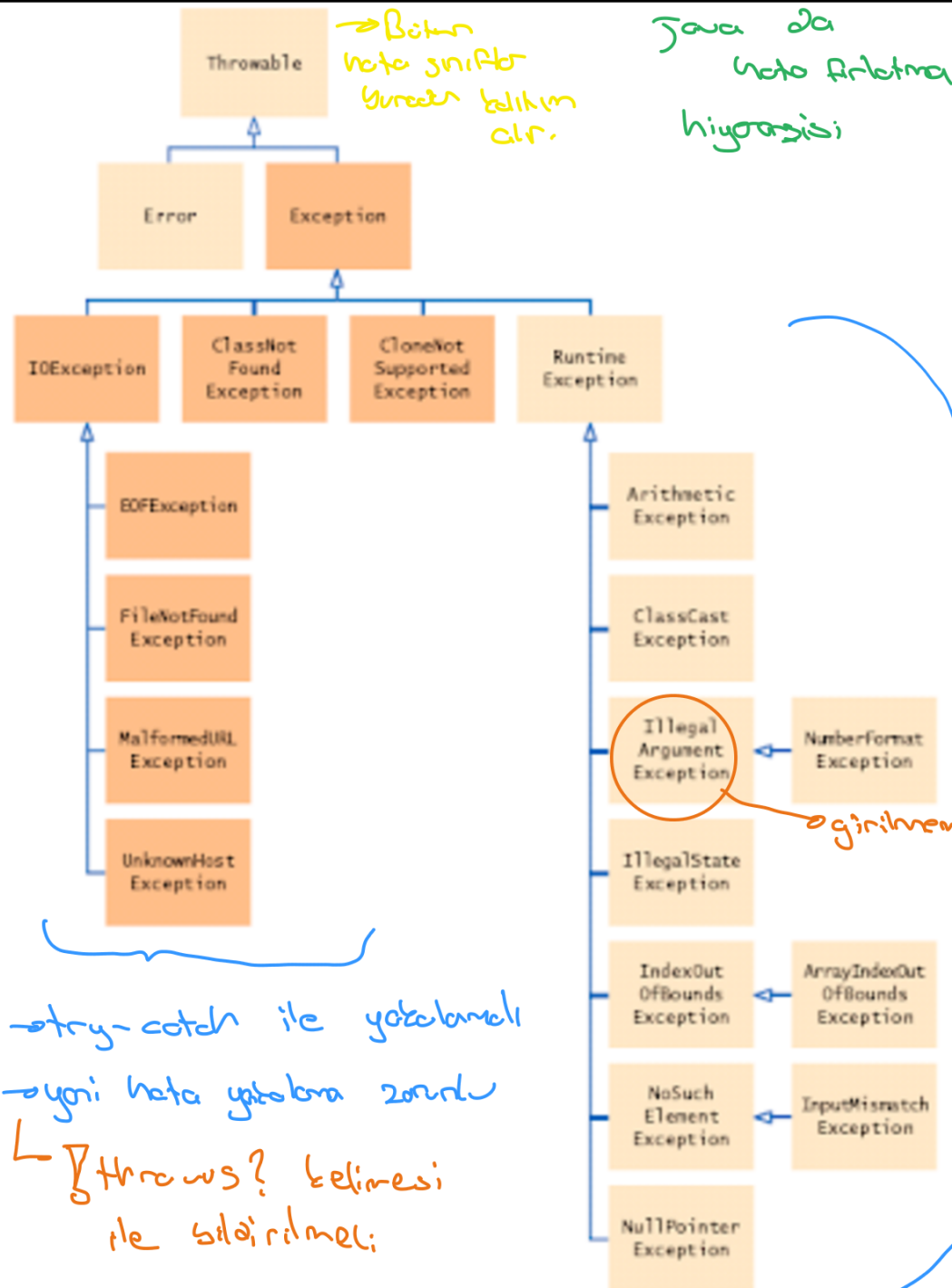
C'de yok.

try-catch yazılır

race? (python) → fırlatır

Java

→ Neane fırlatılır.



throw: hatayı fırlatır

try:

istisnai durumun oluşma ihtimali kodu koyulur.

catch:

hangi sınıfın yakalanması bildirilir.

Figure 1 The Hierarchy of Exception Classes

Or

```

throw new IllegalArgumentException("etgi bakiye");
try {
    // hata yoktur ve Arithmetik.
    x = cin.nextInt(32);
    hesap.paraCek(x);
    // hata olursa ihtimali olan tam
    // try içine yazılır. Hata oluşursa
    // ilgili catch çalışacaktır.
} catch (IllegalArgumentException ex) {
    // beklenen hata belirtilmeli:
    System.out.println(ex.getMessage());
    // mesaj ekrana basılır.
}

```

→ C'de jumperlar ile benzetim yapılır.

* setjmp.h kütüphanesi ile kullanılır

```

#include "setjmp.h"

jmp_buf yatakyici;

```

Or

```

int bolme(int x, int y) {
    if (y == 0) longjmp(yatakyici, -3);
    return x/y;
}

int main() {
    int a=10, b=0;
    if (setjmp(yatakyici) == 0) printf("%d", Bolme(a,b));
    else printf("0 hatası");
    return 0;
}

```

Jumper .devrülüşünde, sistemin 2'ye bölünür. Örgün sistem bekle, 2. si devam eder. İkinci yatakyici değişimini -3 olursa örgün sistemin bu satırı geçemez. İşlemler gerçekleştirmez.

printf("%d", Bolme(a,b));
 catch bunu else ile gerçekleştirir

