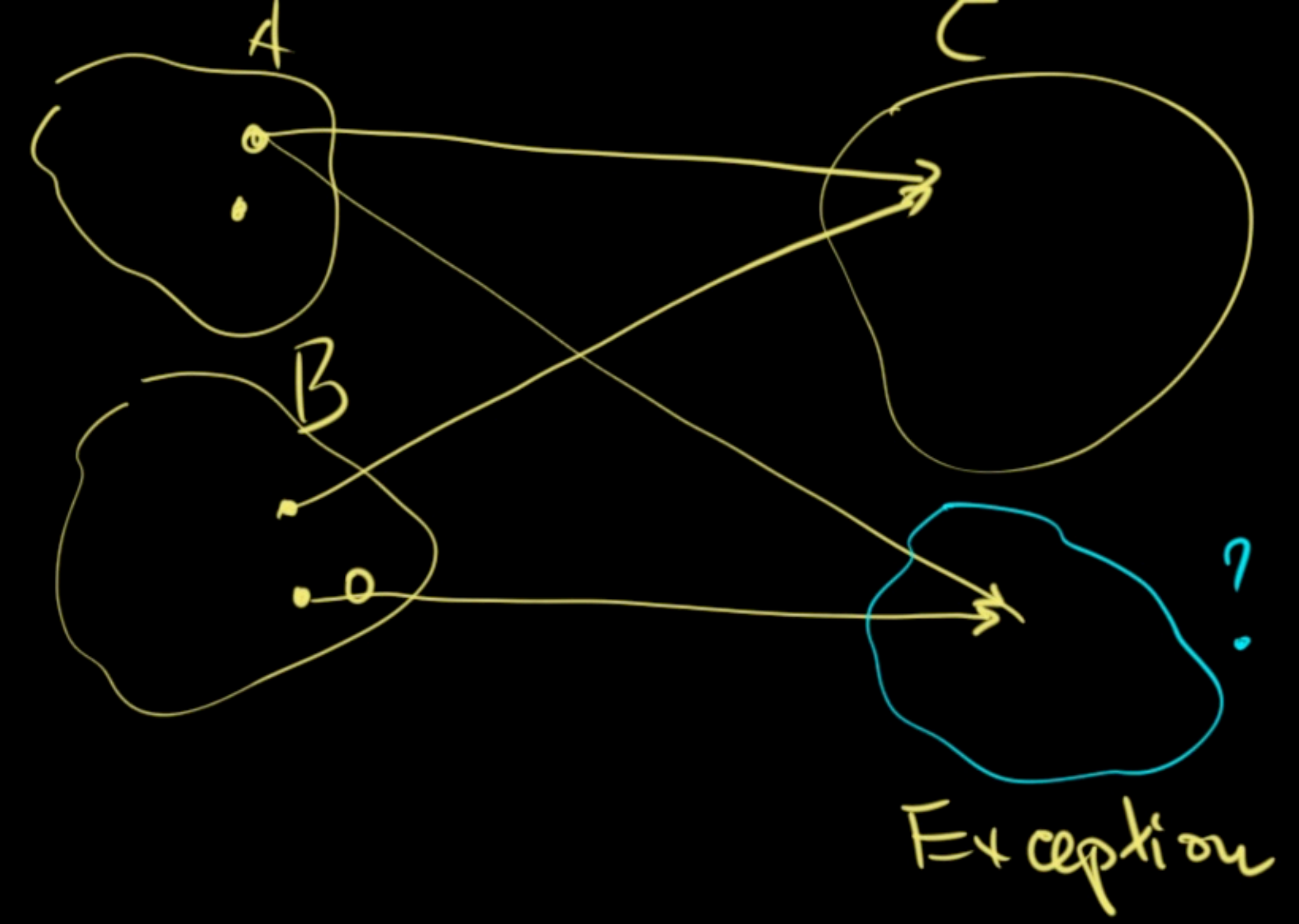
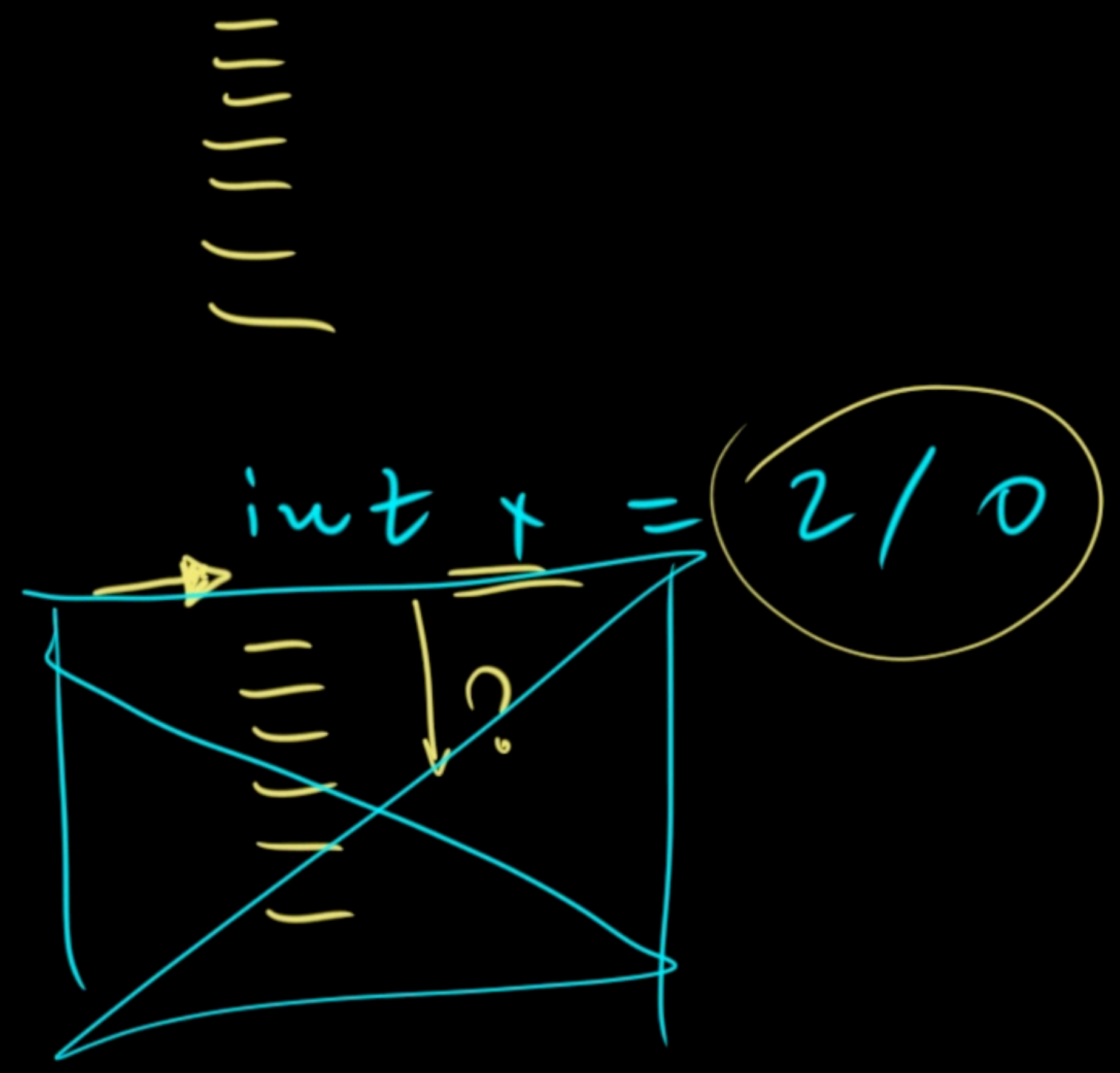


^C
int x = 2
^A
0
_B



Integer.parseInt (String) → Int

str → Int

str → ?
Exception

"123" → 123

"a" → ?

int max(int[]) ← no way to implement

[1, 3, 2]

→ 3

[1]

→ 1

[]

→

0?

[0]

→

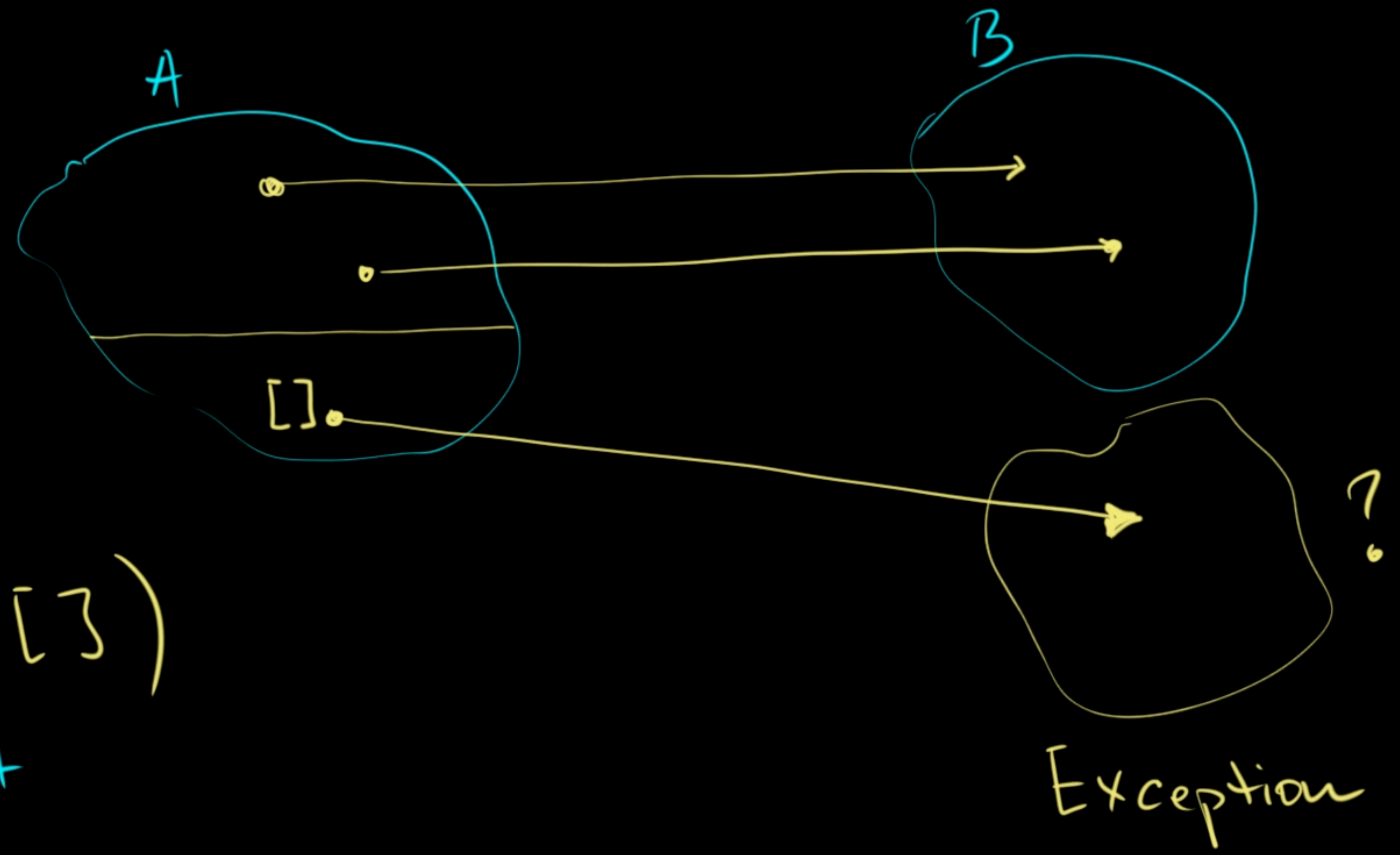
0

[-5, 0]

→

0

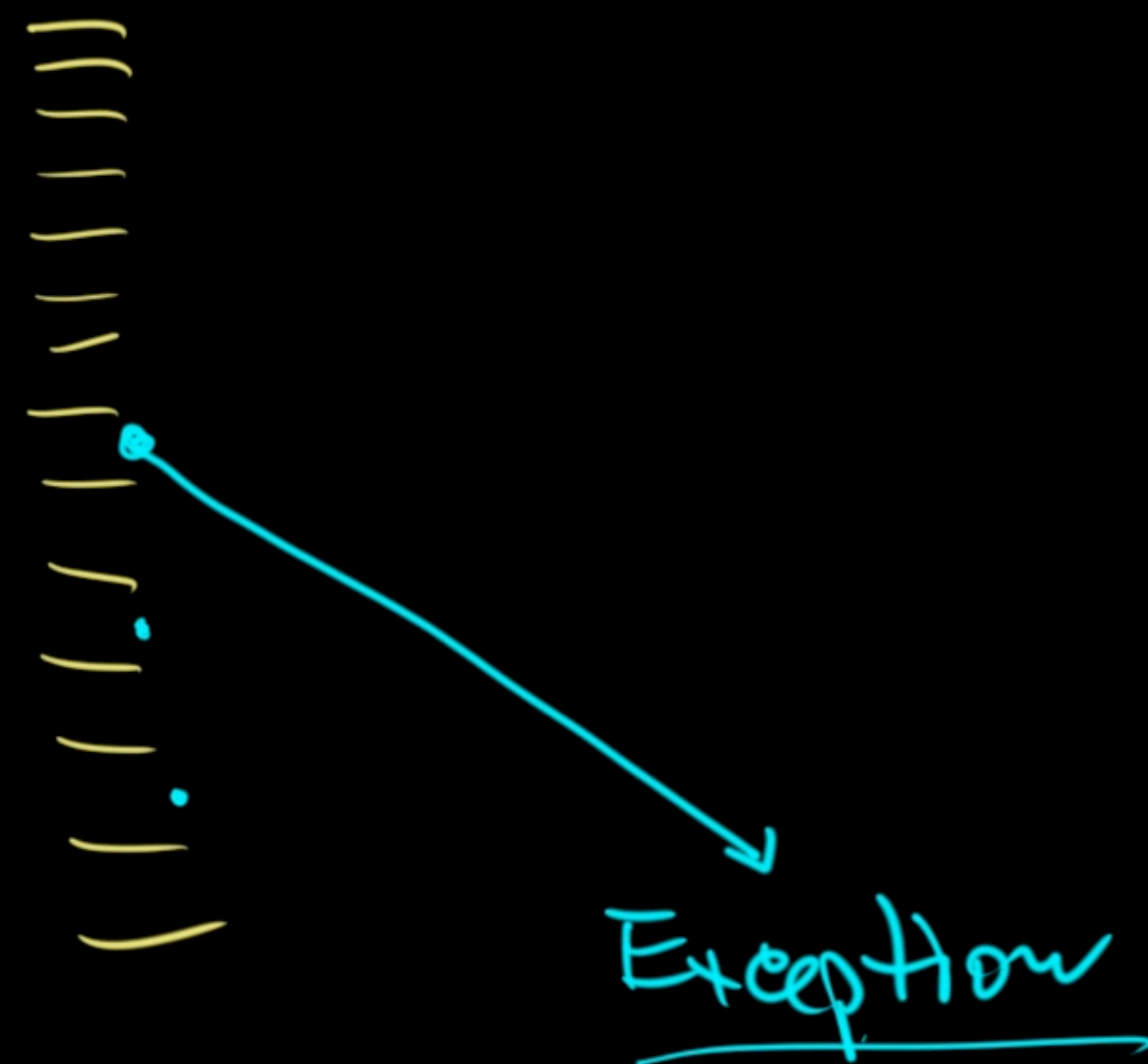
$f: A \rightarrow B$



$\text{int } \max(\text{int}[])$
B A


```
int a = 1;  
int b = 0;  
int c = a / b;  
System.out.printf("hello %d", c);
```

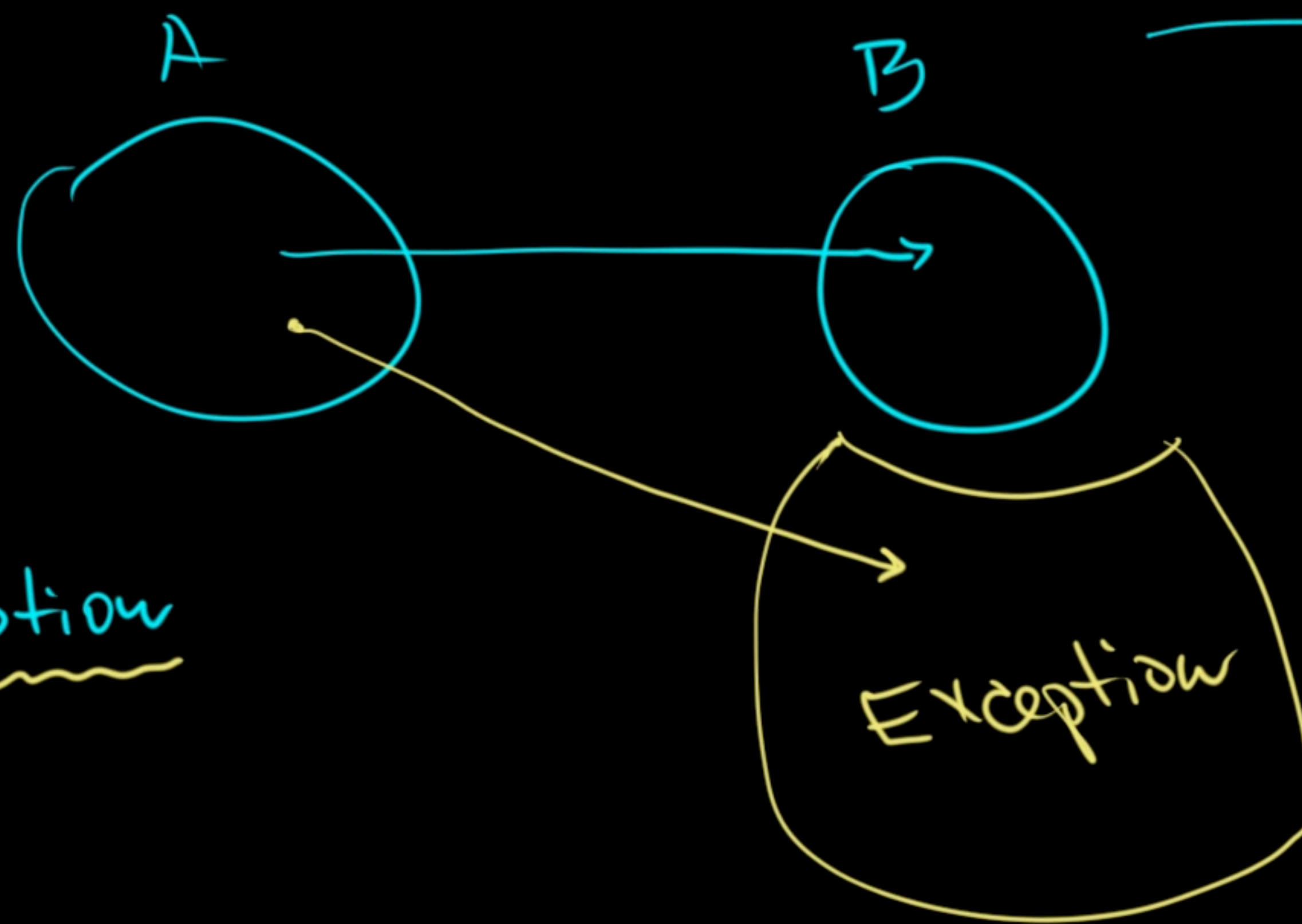
problem



$f: A \rightarrow B$



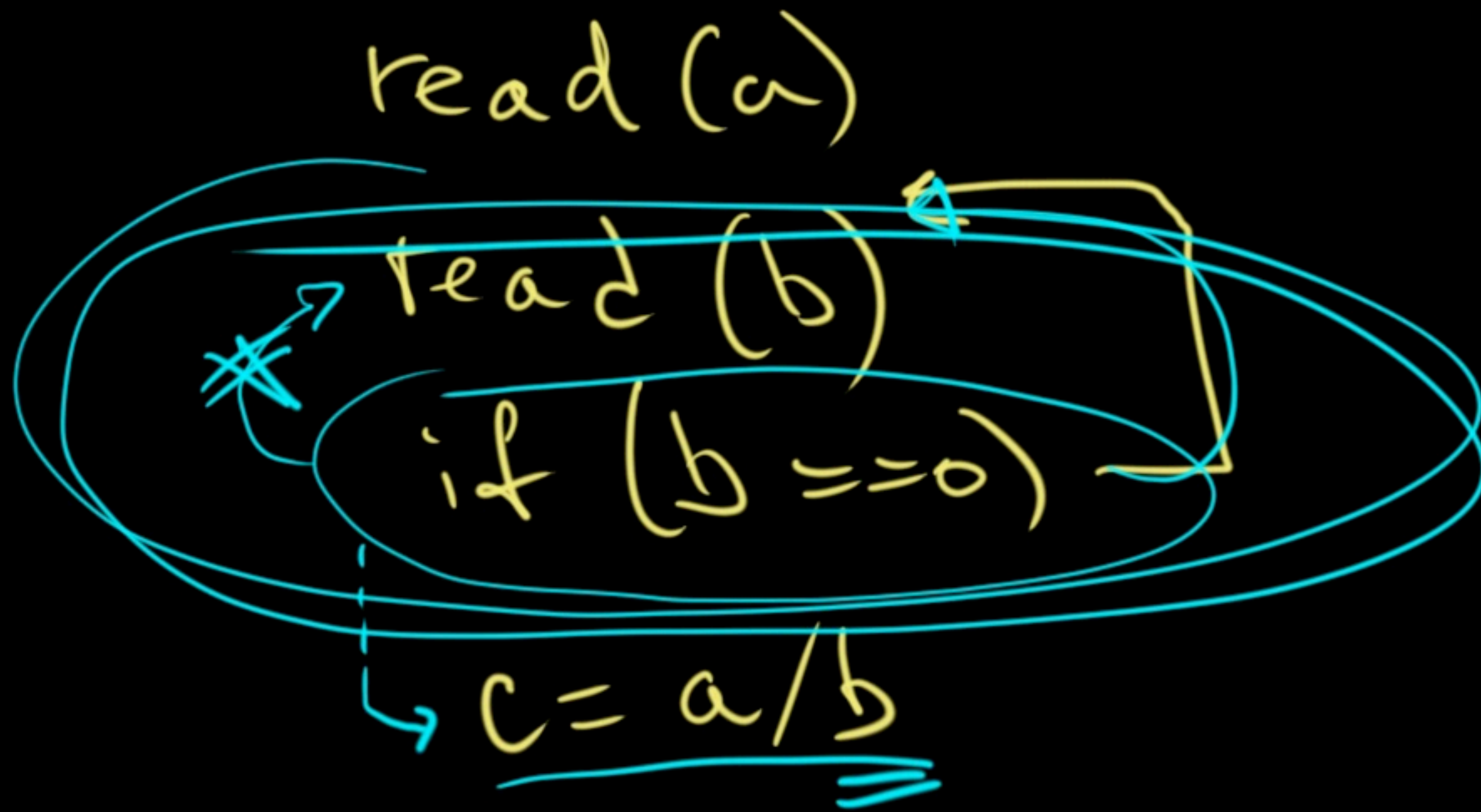
$f: A \rightarrow B | \text{Exception}$



```
int a = 1;  
int b = 0;  
int c = a / b;  
System.out.printf("hello %d", c);
```

if (b == 0)

$f: A \rightarrow B$



void whatever (a,b) {

```
int a = 1;
int b = 2;
int c = a / b;
System.out.printf("hello %d", c);
```

}

f(a,b) → void
 ↓
 f(a,b) → void

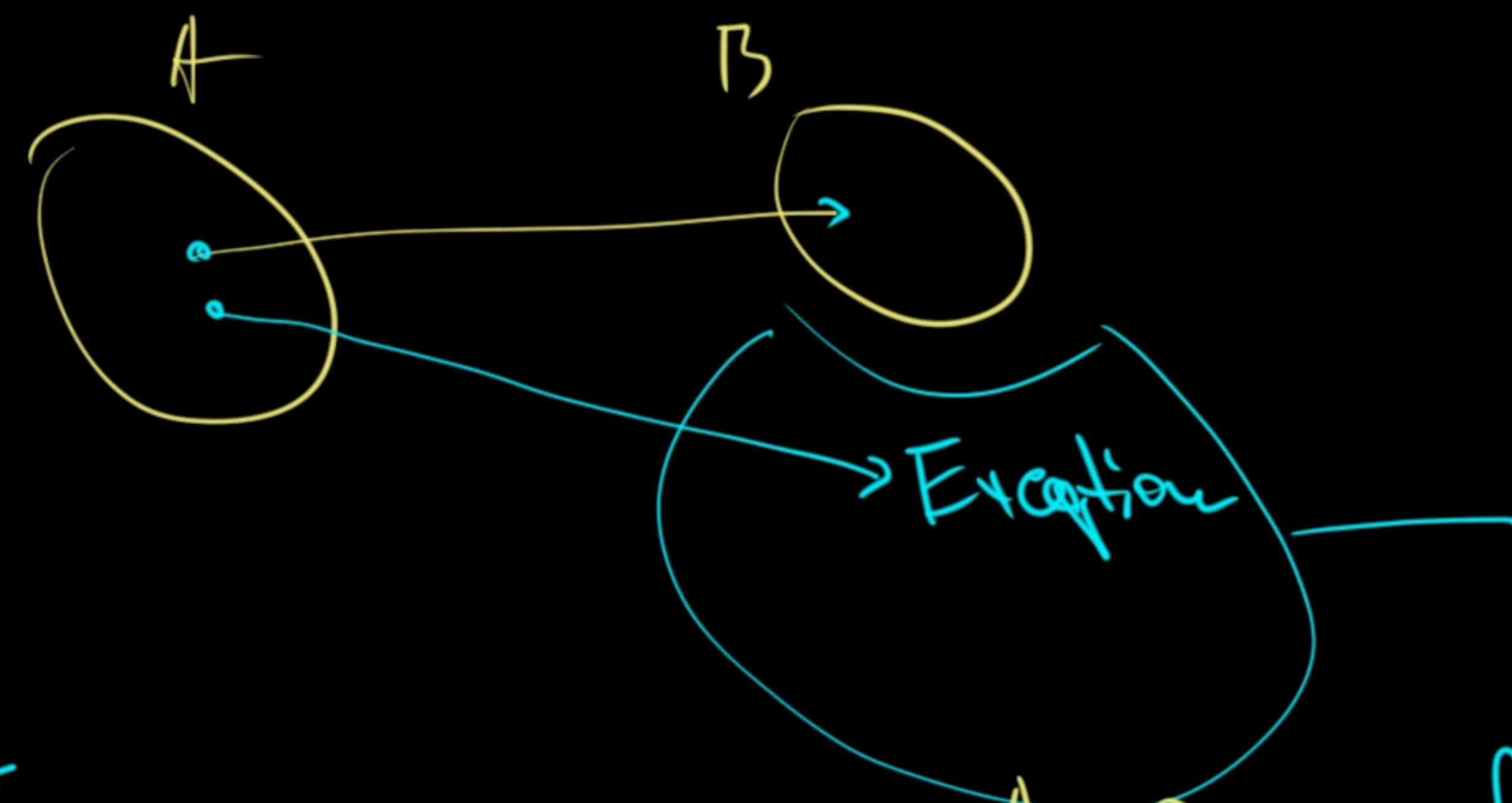
Exception

a = read()
 b = read()

whatever (a,b)

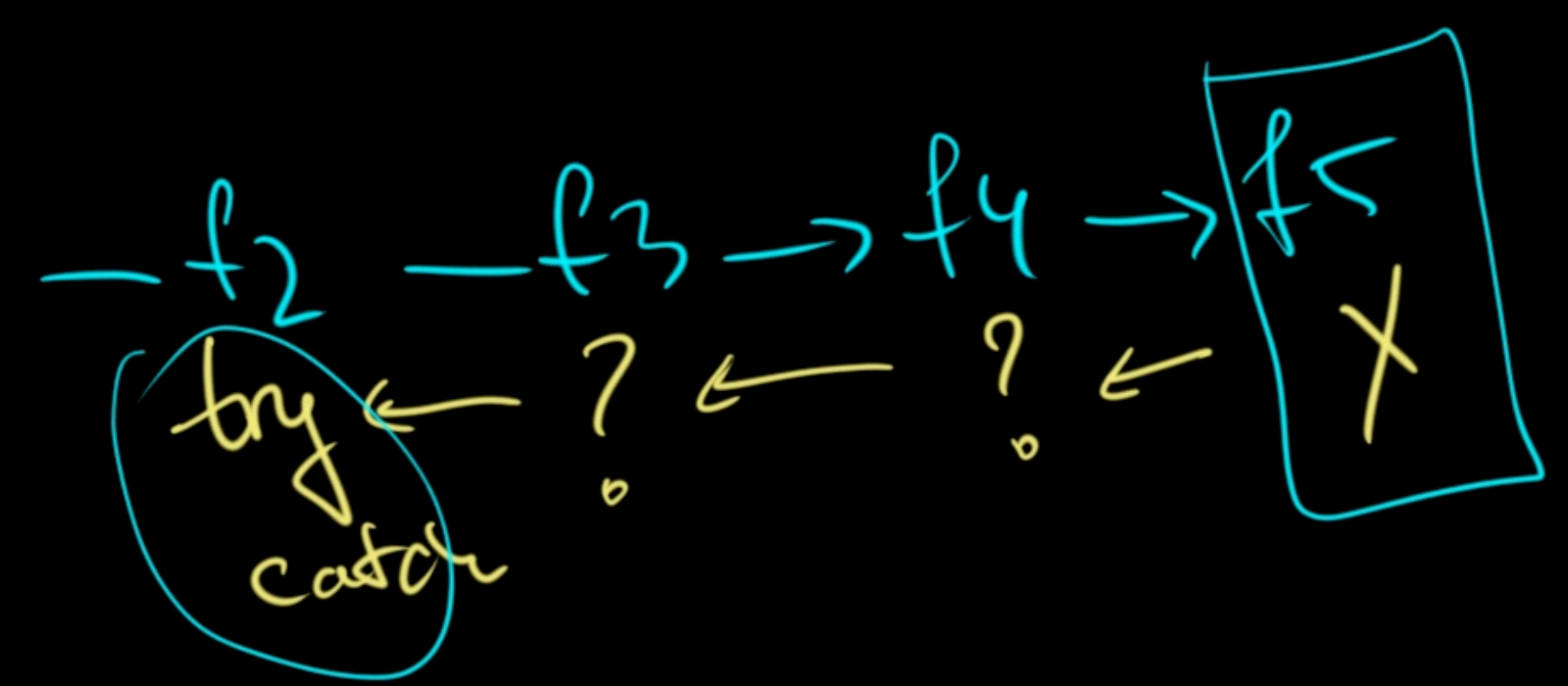
calculator

```
x = read()
y = read()
c = x / y
```

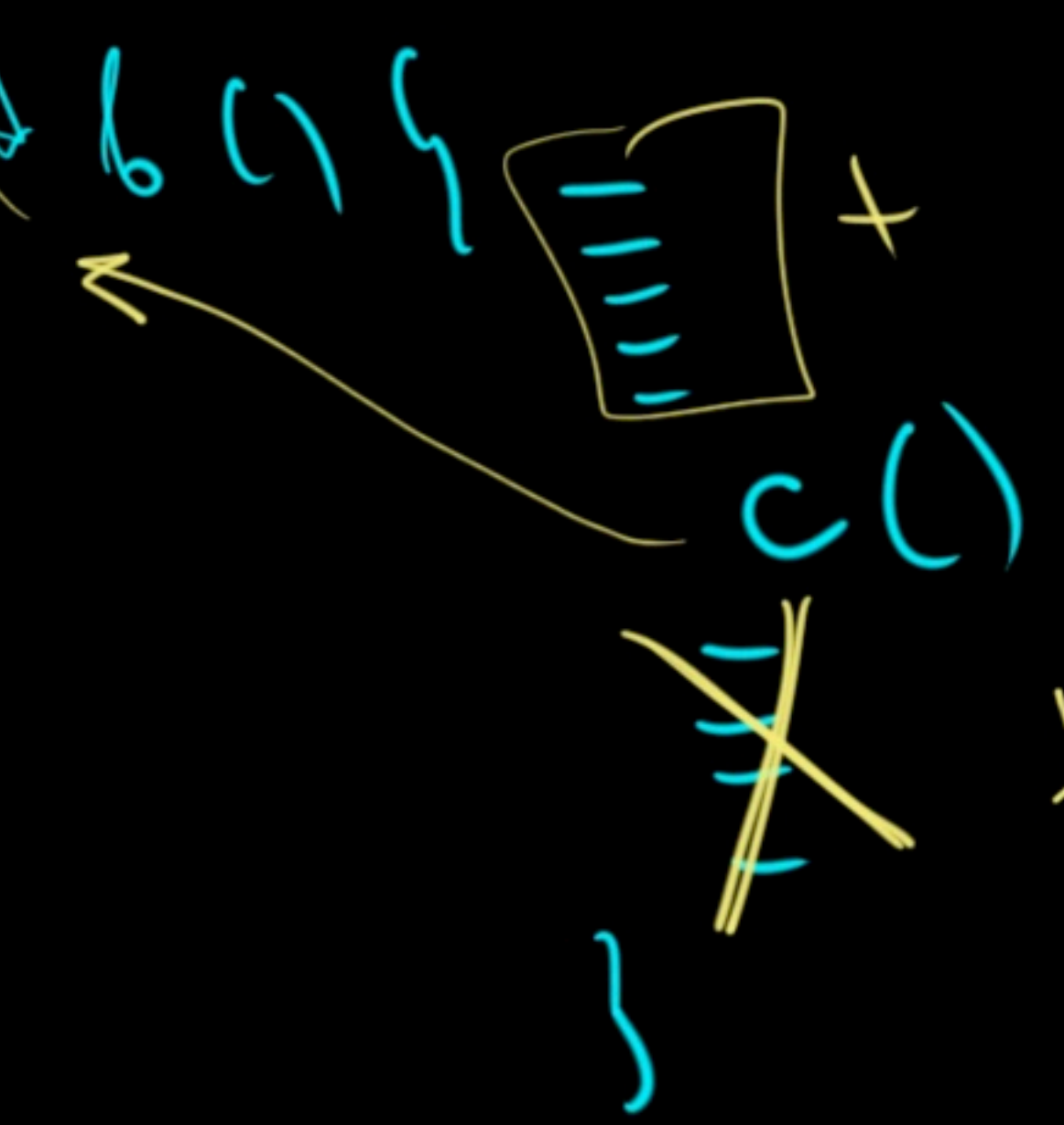



$A \rightarrow B$
 $A \rightarrow B | \text{Exception}$

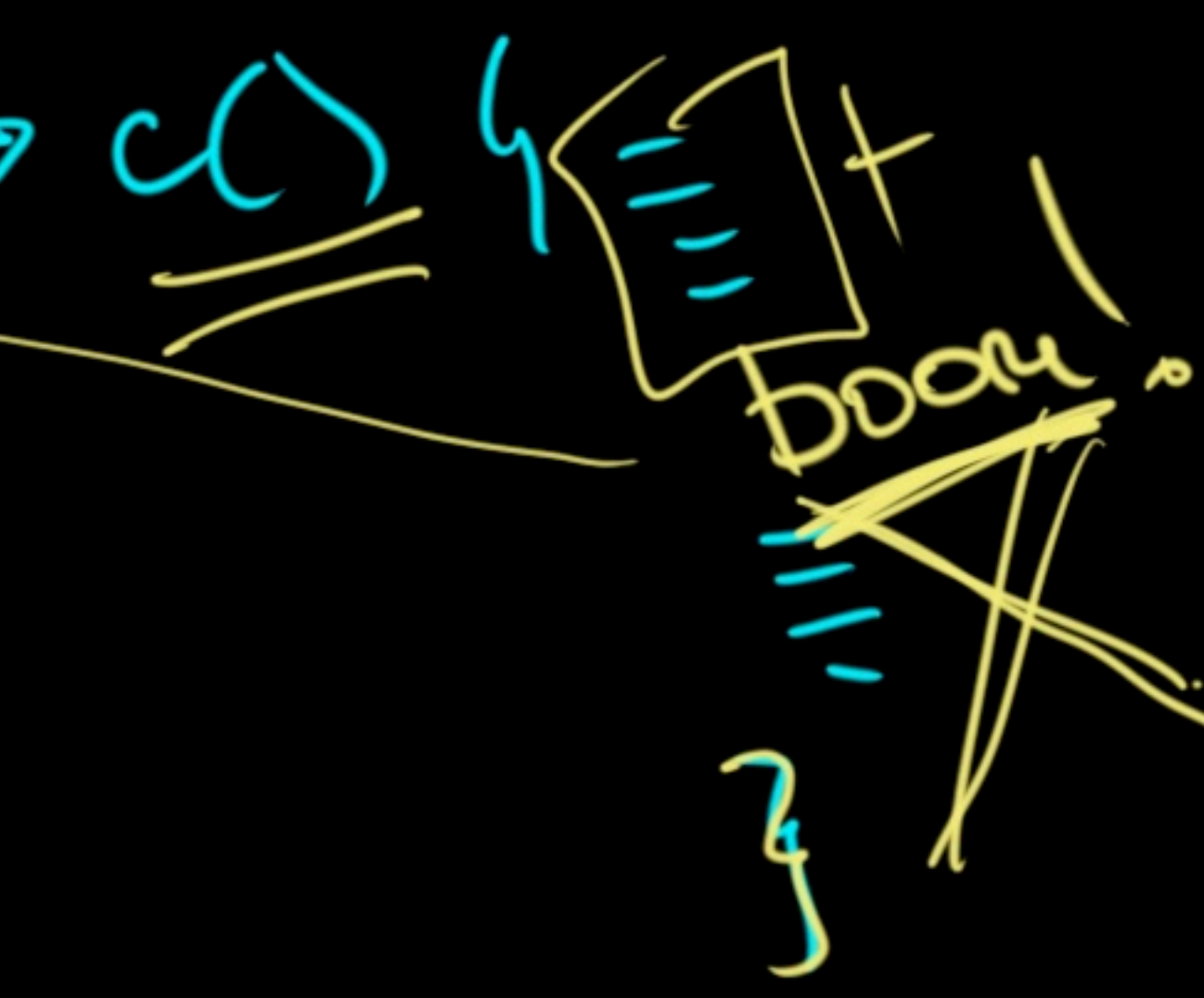
$APP \rightarrow f1$



try/catch



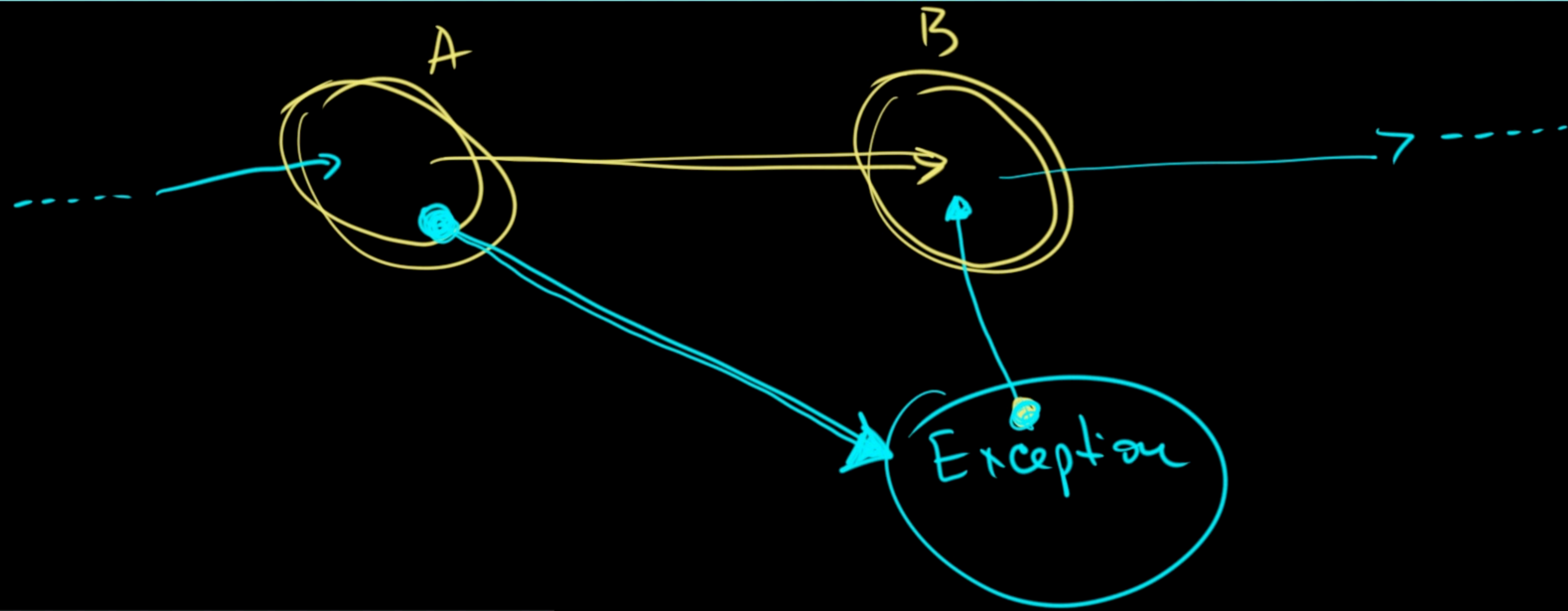
try/catch



catch

$f: A \rightarrow B$

try
catch

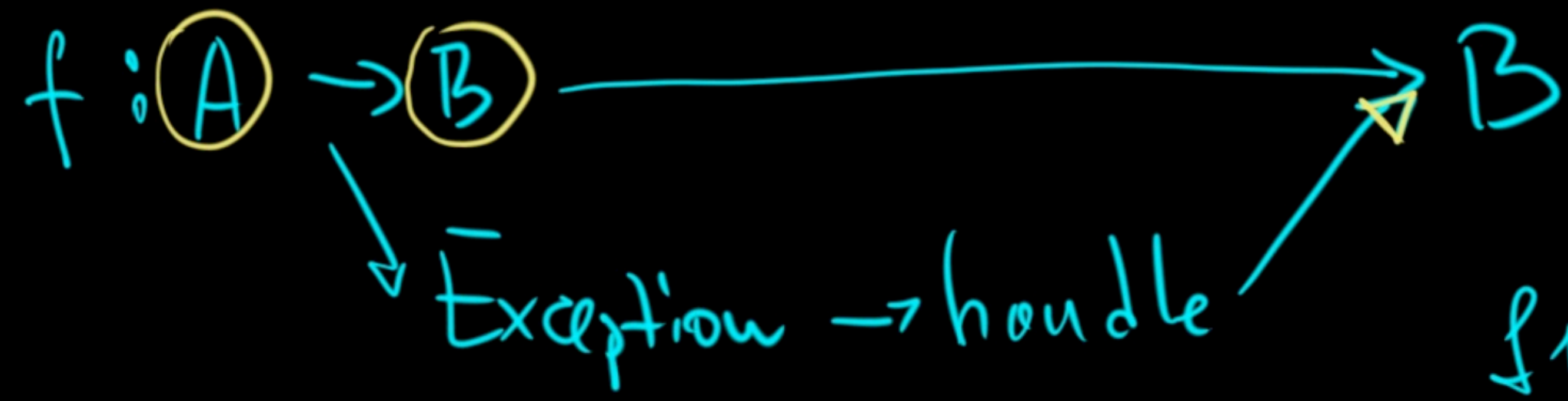


```

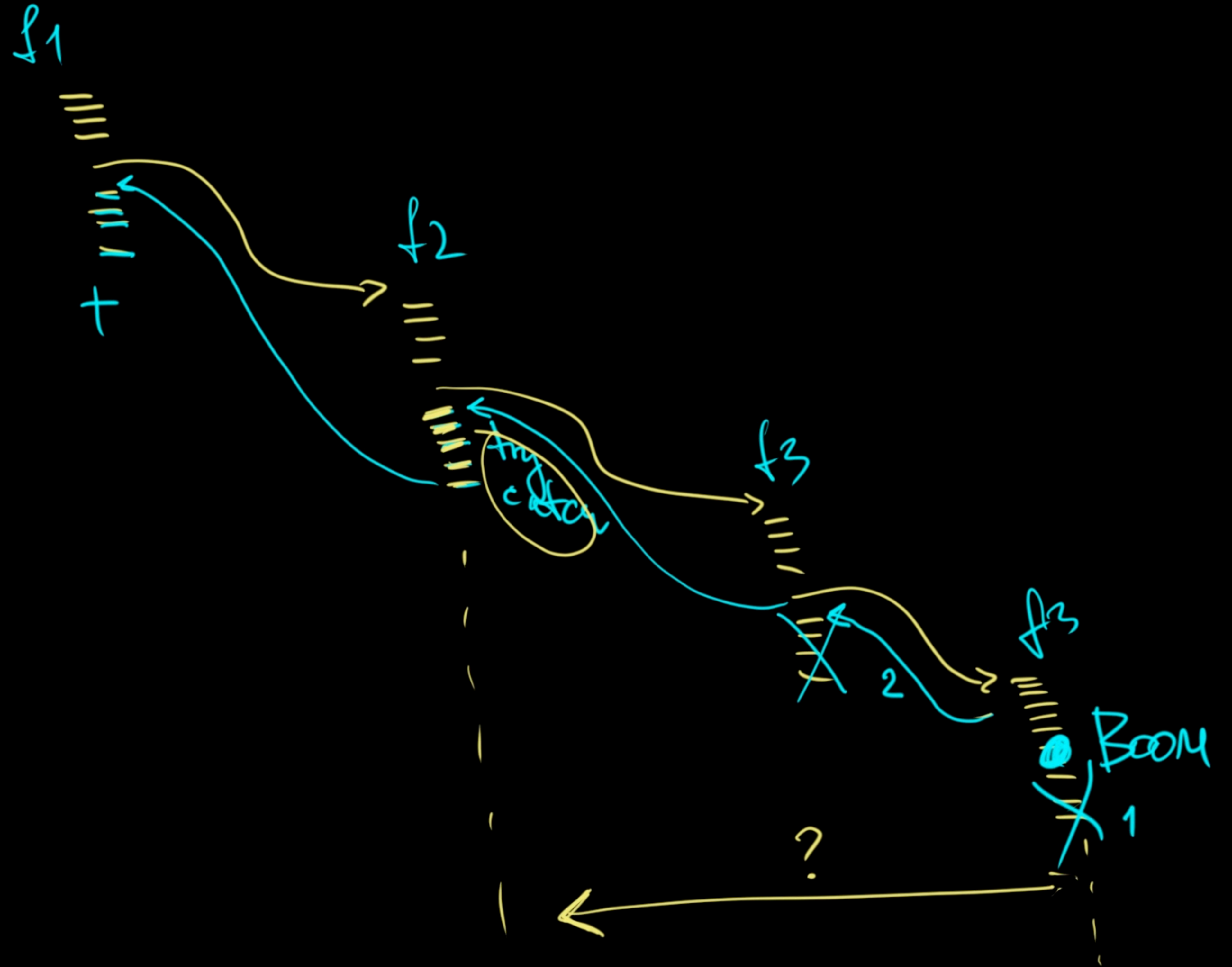
String s1 = sc.nextLine();
String s2 = sc.nextLine();
int a = Integer.parseInt(s1);
int b = Integer.parseInt(s2);

try {
    int c = Calculator.div(a, b);
    System.out.println(c);
} catch (ArithmeticException e) {
    System.out.println("can't divide by zero");
}
  
```

$f_1 \rightarrow f_2 \rightarrow f_3 \rightarrow f_4 \rightarrow$



handle: Exception \rightarrow B



Exception → class

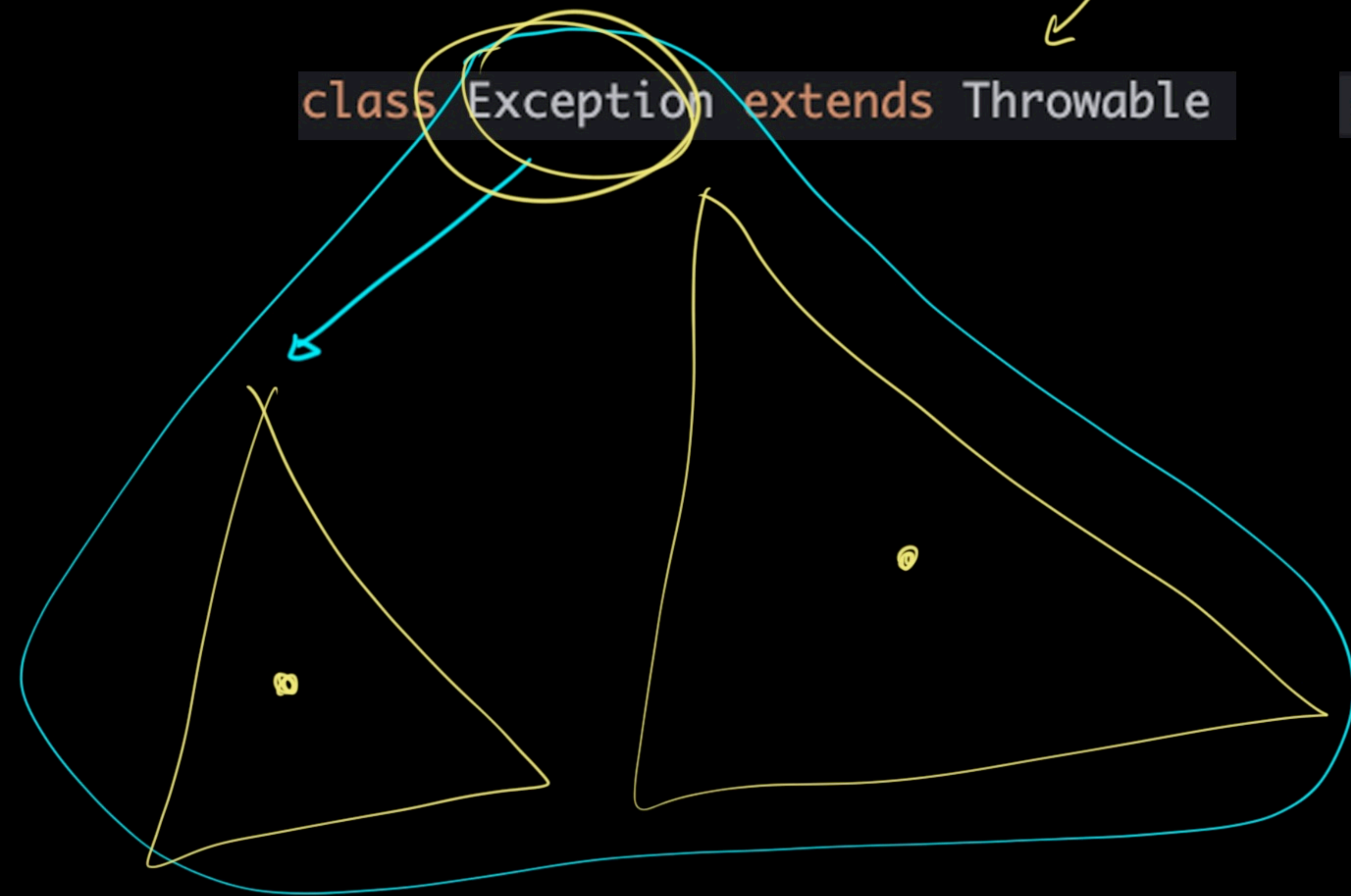
```
throw new NumberFormatException()
```

```
class Throwable
```

```
class Exception extends Throwable
```

```
class Error extends Throwable
```

JVMError StackOverflowError OutOfMemoryError




```
File file = new File(pathname: "1.txt");
var r = new FileReader(file);
```

```
public FileReader( @NotNull File file) throws FileNotFoundException
```

Checked Exception

extends Exception

```
class IOException extends Exception
```

```
class FileNotFoundException extends IOException
```

```
var x = Integer.parseInt(s);
```

```
static int parseInt(String s) throws NumberFormatException
```

non-checked Exception

extends RuntimeException

```
class RuntimeException extends Exception
```

```
class IllegalArgumentException extends RuntimeException
```

```
class NumberFormatException extends IllegalArgumentException
```


Exceptions (738)

2/3 (480)

checked
exceptions

must check →

(260)

unchecked
exceptions

1/3

maybe check
maybe No

```
static void example1(String[] args) {  
    File file = new File("1.txt");  
    try {  
        var r = new FileReader(file);  
    } catch (FileNotFoundException e) {  
        ///  
        ///  
        ///  
    }  
}
```


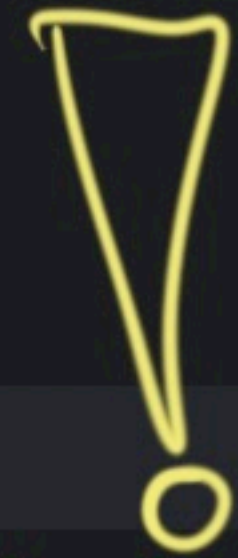
```
static void example2(String[] args) throws FileNotFoundException {  
    File file = new File("1.txt");  
    var r = new FileReader(file);  
}
```

all propagate
to caller level

Checked Ex → NonChecked Ex

Non Checked → Checked Ex

```
try {  
    int c = Calculator2.div(a, b);  
    System.out.println(c);  
} catch (Exception e) {  
    System.out.println("can't divide by zero");  
}
```



```
try {  
    int c = Calculator2.div(a, b);  
    System.out.println(c);  
} catch (ArithmeticException e) {  
    → System.out.println("can't divide by zero");  
} catch (DivisionByZeroException e) {  
    → System.out.println("can't divide by zero !!!");  
}
```


date y m d
time h m s ms us

31.12.2024
12/31/2024
2024-12-31

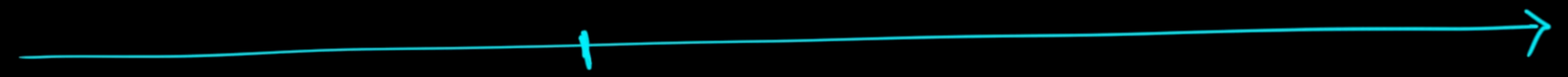
$$d1 \leq d2$$
$$t1 \geq t2$$

15:10 VA
14:10 PL
13:10 England

13.06

data + time

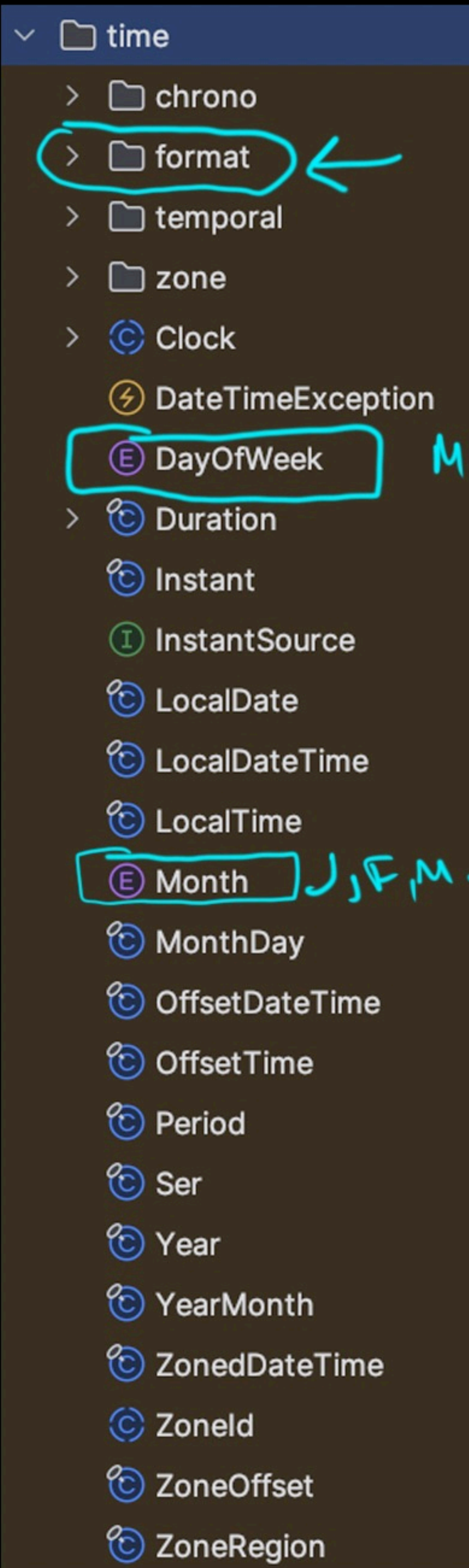
+5:30 India
Gri-Lanka
+5:45 Khatmandu



31.10.2012 07:15 +02:00


```
package java.time;
```

2014

$$\begin{matrix} d_1 \leq d_2 \\ t_1 > t_2 \end{matrix}$$


M, T, W, ...

J, F, M, ... N, D

Date Data+Time
Time

Time zone

Date+Time+Time zone

Instant

Period

2d

parsing

String \rightarrow D
$$\pm 14.10.2024 + 2d + 1M$$

Comparable < >

- isBefore
- isAfter
- equals

D, T \rightarrow String

31.10.2024

10/31/2024

2024-10-31

31-10

DateTimeFormatter

(int, int, int) \rightarrow T

h m s

1) $Str \rightarrow T$

2) $T \rightarrow Str$

3) $(int, int, \dots) \rightarrow T$

4) $T_1 \oplus T_2$

5) $T_1 > / < T_2$

6) $T \rightarrow (int, int, \dots)$

Date
time

+ -

> / <

> / <

> / <

+ -

S → T

S → T



int

+ -
> / <
> / <
> / <