

# Dodatek

## Try..Catch

Być może zauważyłeś że Twój program mógł wyrzucić błąd podczas jego działania. Najłatwiej można to zobaczyć przy `Scanner`.

```
Scanner in = new Scanner(System.in);
int x = in.nextInt();
```

Program działa.. dopóki podamy liczbe na wejściu, ale czy spróbowałeś podać literę? Dostaniesz coś takiego:

```
Exception in thread "main" java.util.InputMismatchException Create breakpoint
    at java.base/java.util.Scanner.throwFor(Scanner.java:939)
    at java.base/java.util.Scanner.next(Scanner.java:1594)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2258)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2212)
    at Main.main(Main.java:16)
```

Jest to tak zwany **wyjątek**, czyli coś co program nie jest w stanie wykonać. W tym przypadku nie jest w stanie zapisać do zmiennej typu `int` tego co mu przekazaliśmy, np. literę `a`.

Kolejnym sposobem na wywołanie **wyjątku**, jest próba wykonania metody na typie złożonym który.. nie został jeszcze stworzony.

```
Scanner in = null; // nie używamy operatora new, czyli nie stworzyliśmy go
jeszcze
int x = in.nextInt();
```

Wyjątek typu `NullPointerException`

```
Exception in thread "main" java.lang.NullPointerException Create breakpoint
    at Main.main(Main.java:16)
```

W takich sytuacjach mówi że *program wyrzucił wyjątek*

`null` to słowo kluczowe dla typów złożonych. Aby stworzyć typ złożony musimy użyć operatora `new` (wyjątkiem od tej reguły są typy opakowujące i typ `String`). Jednak jeżeli nie chcemy tworzyć od razu typu złożonego, możemy użyć słowa kluczowego `null`.

Jak pamiętasz zmienna jest tylko kluczem do pojemnika danych, a poprzez przypisanie `null` do typu złożonego mówimy że ten klucz nie jest jeszcze przypisany do żadnego pojemnika. Mówimy że "wskazuje na nic".

---

No dobrze, tu się rodzi pytanie czy możemy przewidzieć takie sytuacje i czy możemy na nie zareagować?

Możemy, do tego celu służy konstrukcja `try..catch`. Wygląda ona tak

```
try{
    // Ryzykowny kod programu
}catch(Exception e){ // Typ złożony 'Exception', dostarcza nam informacji o
    błędzie.
    // w to miejsce przejdzie kiedy program 'wyrzuci wyjątek'
}
```

Przykład

```
Scanner in = new Scanner(System.in);
int x;
try{
    x = in.nextInt();
}catch(Exception e){
    // kiedy podamy np. 'a' na wejściu
    System.out.println("Zła wartość! To nie liczba!");
    return;
}
```

to tyle :)

.  
.  
.  
.

**Tak na prawdę to nie** ale na ten moment wystarczy Ci poznanie podstawowej składni :)