

Wyjątki

Wyjątki to sytuacje nieprzewidziane, które mogą się zdarzyć podczas obsługi aplikacji. Każdy wyjątek dziedziczy po **Throwable**. Rodzaje wyjątków:

1. **Checked exceptions** – muszą być obsługiwane przez programistę, np.: `IOException`
2. **Unchecked exceptions** (tzw **runtime exception**) – te wyjątki się naprawia w kodzie, a nie wyłapuje.

Obsługa wyjątków

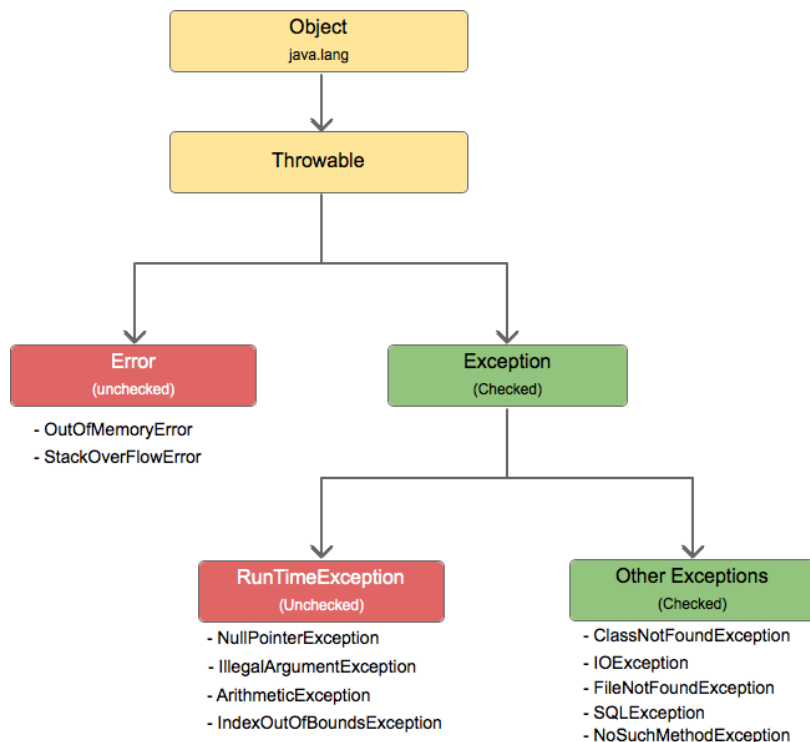
Mamy dwa rodzaje obsługi wyjątków:

1. **try, catch**(nie musi być jeśli jest **finally**), **finally**(zawsze wykonywany)
2. **throws**

W przypadku użycia **try-catch** pierwszy znaleziony pasujący typ wyjątku w **catch** powoduje, że kolejne nie są już sprawdzane. Można jednak napisać alternatywę klas wyjątków, które powinny być wyłapane

```
catch (ArithmeticException | IllegalArgumentException exception)
```

Na wyjątki typu **Exception** nie mamy wpływu i musimy je obsługiwać. (np. brak połączenia z netem)



Wyjątki **Error** może rzucać tylko JVM. np. `StackOverflowError`.

Obsługa wyjątków jest bardzo pamięciożerna i powolna. Jeśli chodzi o własne wyjątki, to możemy rzucać tylko wyjątkami RuntimeException.

Własny wyjątek

<https://github.com/idzikpro/JavaBasics/blob/master/src/main/java/pl/idzikpro/exception/ExceptionSumOver.java>

```
public class ExceptionSumOver extends Exception {  
  
    public ExceptionSumOver() {  
        super();  
    }  
  
    public ExceptionSumOver(String message) {  
        super(message);  
    }  
}
```

```
int a = 5;  
int b = 6;  
try {  
    int sum = a + b;  
    if (sum > 10) throw new ExceptionSumOver("sum over 10");  
} catch (ExceptionSumOver e) {  
    System.out.println(e.getMessage());  
}
```