

Mechanizm wyjątków, który z grubsza [opisałem wcześniej](#). Wyjątek jest klasą, a (prawie) po każdej klasie można dziedziczyć. Po co dziedziczyć po wyjątkach? A no właśnie - tego się dowiesz za chwilę.

## Dziedziczenie

Aby rozszerzyć (dziedziczyć) jakąś klasę używamy słowa kluczowego `extends`:

```
{codecitation class='brush: php'}class MyException extends Exception {}{/codecitation}
```

Aby rzucić nasz własny wyjątek używamy składni:

```
{codecitation class='brush: php'}throw new MyException('Treść komunikatu');{/codecitation}
```

## Wiele bloków catch

Skoro jest możliwość rzucenia wielu typów wyjątków, to i warto byłoby móc przechwycić różnego rodzaju wyjątki. Można:

```
{codecitation class='brush: php'}  
  
try {    throw new MyException('testowy wyjątek'); } catch(MyException $e) {    echo  
'Przechwyciłem MyException'; } catch(Exception $e) {    echo 'Przechwyciłem Exception'; }  
  
{/codecitation}
```

Działa to podobnie do switcha. Podajemy klasę wyjątku i obsługę dla tego wyjątku. Gdyby

## Hierarchia klas wyjątków

Wpisany przez Patryk yarpo Jar  
wtorek, 27 lipca 2010 22:27

---

zamienić kolejność w powyższym przykładzie, to przechwycony byłby wyjątek klasy Exception, ponieważ (dziedziczenie) wyjątek klasy MyException jest jednocześnie widziany jako wyjątek klasy Exception. Pseudomatematycznie można to zapisać: MyException > Exception, czyli "rozszerza".

## Po co?

Dzięki odpowiedniej hierarchii klas wyjątków możemy stworzyć różne wyjątki na różne niechciane, ale mogące się zdarzyć, sytuacje. A następnie w czytelny sposób je rozpoznać i obsłużyć:

```
{codecitation class='brush: php'}
```

```
try { $oSys = new System(); $oSys->work(); } catch(NoFileException $e) { echo 'Przechwyciłem NoFileException'; } catch(PDOException $e) { echo 'Przechwyciłem PDOException'; } catch(ArrayException $e) { echo 'Przechwyciłem ArrayException'; } catch(Exception $e) { echo 'Przechwyciłem Exception'; }
```

```
{/codecitation}
```

Zamiast echo w takim wypadku można użyć np. funkcji [mail](#) (lub [obudowanej w klasę](#)) wysyłając sobie na maila informację co się stało. Bardzo przydatna może być wtedy taka konstrukcja:

```
{codecitation class='brush: php'}
```

```
catch(Exception $e) { mail('a(at)b.pl', 'błąd na stronie', var_export($e, true)); }
```

```
{/codecitation}
```

## Hierarchia klas wyjątków

Wpisany przez Patryk yarpo Jar  
wtorek, 27 lipca 2010 22:27

---

Funkcja ``var_export`` rozłoży nam na części pierwsze wszystkie informacje o wyjątku. Na maila dostaniemy rozpiskę co się stało, w której linii, jakiej funkcji, z jakimi parametrami.

## Co dalej?

Zapraszam do lektury artykułu o [walidacji danych w klasie wyjątku](#) .