

## Исключения

Модель исключений (exceptions) в PHP 5 схожа с используемыми в других языках программирования. Исключение можно сгенерировать (как говорят, "выбросить") при помощи оператора `throw`, и можно перехватить (или, как говорят, "поймать") оператором `catch`. Код генерирующий исключение, должен быть окружен блоком `try`, для того чтобы можно было перехватить исключение. Каждый блок `try` должен иметь как минимум один соответствующий ему блок `catch` или `finally`.

Генерируемый объект должен принадлежать классу `Exception` или наследоваться от `Exception`. Попытка сгенерировать исключение другого класса приведет к неисправимой ошибке.

### catch

Можно использовать несколько блоков `catch`, перехватывающих различные классы исключений. Нормальное выполнение (когда не генерируются исключения в блоках `try`) будет продолжено за последним блоком `catch`. Исключения так же могут быть сгенерированы (или вызваны еще раз) оператором `throw` внутри блока `catch`.

### finally

В PHP можно использовать блок `finally` после или вместо блока `catch`. Код в блоке `finally` всегда будет выполняться после кода в блоках `try` и `catch`, вне зависимости было ли брошено исключение или нет, перед тем как продолжится нормальное выполнение кода.

```
<?php
function inverse($x) {
    if (!$x) {
        throw new Exception('Division by zero.');
```

```
    }
    return 1/$x;
}

try {
    echo inverse(5); // 0.2
    echo inverse(0); // Exception: Division by zero.
} catch (Exception $e) {
    echo 'Exception: ', $e->getMessage();
}

?>
```

```
<?php
function inverse($x) {
    if (!$x) {
        throw new Exception('Division by zero.');
```

*→*

```
    }
    return 1/$x;
}

try {
    echo inverse(5);
} catch (Exception $e) {
    echo 'Exception: ', $e->getMessage();
} finally {
    echo "First finally.;"
```

*→*

```
}
```

*→*

```
try {
    echo inverse(0);
} catch (Exception $e) {
    echo 'Exception: ', $e->getMessage();
} finally {
    echo "Second finally.;"
```

*→*

```
}
```

*→*

```
/*
0.2
First finally.
Exception: Division by zero.
Second finally.
*/
?>
```