## Три ключевых понятия ООП

**Аннотация** Их еще называют тремя китами ООП. На этих базовых принципах держится всё объектно-ориентированное программирование. Инкапсуляция, наследование и полиморфизм.

**Ключевые слова** ООП, объектно-ориентированное программирование, объекты, классы, инкапсуляция, наследование, полиморфизм.

**Авторы** Пятосин Андрей Владимирович, Марков Алексей Николаевич, Боровская Ольга Олеговна, Михалькевич Александр Викторович, Писарчик Андрей Юрьевич.

Рассмотрим три ключевых понятия объектно-ориентированного программирования.

**Инкапсуляция** — свойство языка программирования, позволяющее объединить и защитить данные и код в объект и скрыть реализацию объекта от пользователя (прикладного программиста). При этом пользователю предоставляется только спецификация (интерфейс) объекта.

**Наследование** — важнейший механизм объектно-ориентированного программирования, позволяющий описать новый класс на основе уже существующего (родительского), при этом свойства и функциональность родительского класса заимствуются новым классом.

Для наследования классов в PHP используется ключевое слово extends. Пример:

class Article

{

… // содержимое класса

}

class NewsArticle extends Article

{

… // содержимое класса

}

Классы-наследники могут переопределять методы классов-родителей. Пример:

class Article

{

…

// Функция для вывода статьи

function view()

{

echo "<h1>$this->title</h1><p>$this->content</p>";

}

}

class NewsArticle extends Article

{

…

// Функция для вывода статьи

function view()

{

echo "<h1>$this->title</h1><span style='color: red'>".

strftime('%d.%m.%y', $this->datetime).

" <b>Новость</b></span><p>$this->content</p>";

}

}

В данном примере дочерний класс NewsArticle переопределяет метод view родительского класса Article.

### Полиморфизм— взаимозаменяемость объектов с одинаковым интерфейсом.

Язык программирования поддерживает полиморфизм, если классы с одинаковой спецификацией могут иметь различную реализацию— например, реализация класса может быть изменена в процессе наследования.

Пример полиморфизма:

class A

{

function Test() { echo 'Это класс A<br />'; }

function Call() { $this->Test(); }

}

class B extends A

{

function Test() { echo 'Это класс B<br />'; }

}

$a = new A();

$b = new B();

$a->Call(); // выводит: «Это класс A»

$b->Test(); // выводит: «Это класс B»

$b->Call(); // выводит: «Это класс B»