PHP Уровень 2

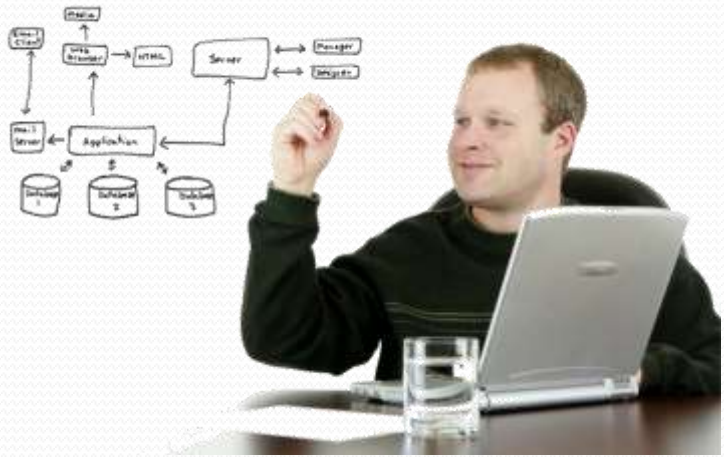
Урок 2

«Объектно-ориентированное программирование»

Объектно-ориентированное программирование

Краткое содержание:

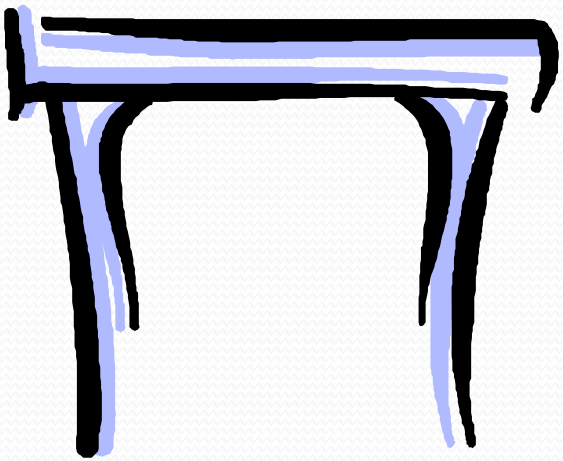
- Парадигма ООП
- Основные понятия ООП
- Три «кита» ООП
- Реализация ООП в PHP



Объектно-ориентированное программирование

Класс

- Класс – это тип данных
- Он описывает, какими будут объекты этого класса
- Класс – это «идея» объектов



Объект

- Объект – это экземпляр класса, реализация идеи, заложенной в классе
- Его структура и поведение определяются классом, которому принадлежит объект

Объектно-ориентированное программирование

Свойства

- **Поля или свойства – переменные, определенные в классе**
- Каждый объект этого класса получает эти переменные
- Они отвечают за свойства объектов:
  - число ножек стола
  - материал
  - цвет и т. д.

Методы

- **Методы – это функции, определенные в классе**
- Они отвечают за поведение объектов класса
- Или за операции, которые можно проводить над объектами:
  - передвинуть
  - накрыть скатертью
  - посчитать количество человек за столом

Объектно-ориентированное программирование

Пример (новостная статья)

```
class Article {
  public $title;
  public $text;
  public function view() {
    echo '<h1>' . $this->title . '</h1>';
    echo '<div>' . $this->text . '</div>';
  }
}
// Два публичных свойства
   и один публичный метод
```

Объектно-ориентированное программирование

Пример (новостная статья)

```
$art = new Article;
$art->title = 'Важная новость!';
$art->text = 'Очень важный текст';
$art->view();
```

- **new** – создание нового объекта класса
- **$this** – «этот», ссылка на объект изнутри его методов

Объектно-ориентированное программирование

Конструктор класса

```
class Article {
  public $title;
  public $text;
  public function __construct($title, $text) {
    $this->title = $title;
    $this->text = $text;
  }
  ...
}

$article = new Article('Заголовок', 'Текст');
$article->view();
```

Объектно-ориентированное программирование

Конструктор класса

- Имеет зарезервированное имя `__construct()`
- Вызывается автоматически при создании объекта данного класса
- Может принимать аргументы, как любой другой метод
- Применяется для начальной установки свойств объекта

Три «кита» ООП

Наследование

- Класс может быть унаследован от другого класса
- Дочерний класс автоматически будет иметь те же свойства и методы, что и родительский
- Класс может иметь сколько угодно наследников, но только одного предка

Три «кита» ООП

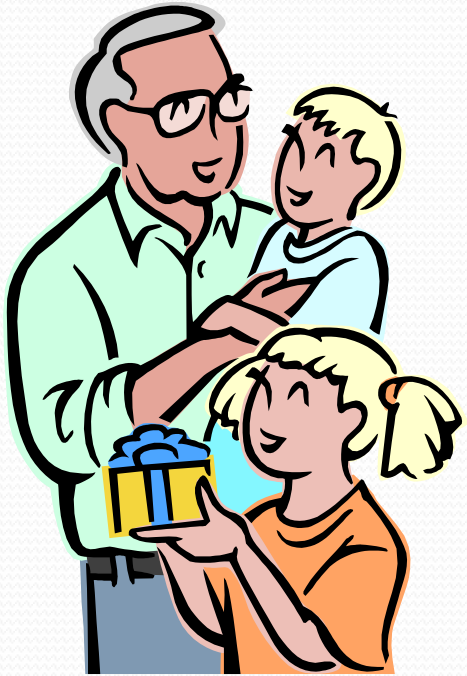
Наследование

```
class Article {
  public $title;
  public $text;
...
}

class NewsArticle extends Article {
  public $source;
}

class RepostArticle extends Article {
  public $source;
  public function __construct() {
    parent::__construct();
    ...
  }
}
```

«**extends**» - наследуется от «**parent**» - ссылка на родительский класс



Три «кита» ООП

Полиморфизм

```
class A {
  public function Test() {
    echo 'Это A';
  }
  function Call() {
    $this->Test();
  }
}

class B extends A {
  public function Test() {
    echo 'Это B';
  }
}
```

Дочерний класс может переопределить методы, которые он получил в наследство от родительского

```
$a = new A();
$b = new B();
$a->Call(); // «Это A»
$b->Test(); // «Это B»
$b->Call(); // «Это B»
```

Три «кита» ООП

Инкапсуляция

- Свойство может быть
  - доступным извне: **public**
  - доступным только в этом классе: **private**
  - доступным в классе и всех его наследника: **protected**
- Такие же модификаторы доступа могут иметь и методы класса

Статические свойства и методы

Константы класса

- Задаются один раз
- Не могут изменяться
- Принадлежат классу, а не объектам

```
class Math {
   const PI = 3.14159;
}
echo Math::PI;
```

Статические свойства и методы

- Принадлежат классу, а не объектам
- Задаются с помощью ключевого слова static
- Внутри класса доступны через self::$prop или self::method()
- Вне класса – через Class::$prop или Class::method()

Абстрактные свойства и методы

Абстрактные классы

- Нельзя создать объект абстрактного класса
- Служит для наследования от него конкретных классов
- Задается ключевым слово **abstract** class

Абстрактные методы

- Не могут содержать тело метода
- Обязаны быть реализованы в дочерних классах
- Задаются ключевым словом **abstract** function

Домашнее задание

- Создайте класс, работающий с базой данных
  - В его конструкторе – подключайтесь к БД
  - Его методы пусть умеют добавлять новую запись в таблицу, обновлять существующую и получать список записей
- Создайте абстрактный класс статьи
- От него унаследуйте класс новости
- Перепишите предыдущее задание, используя классы новостей и базы данных