

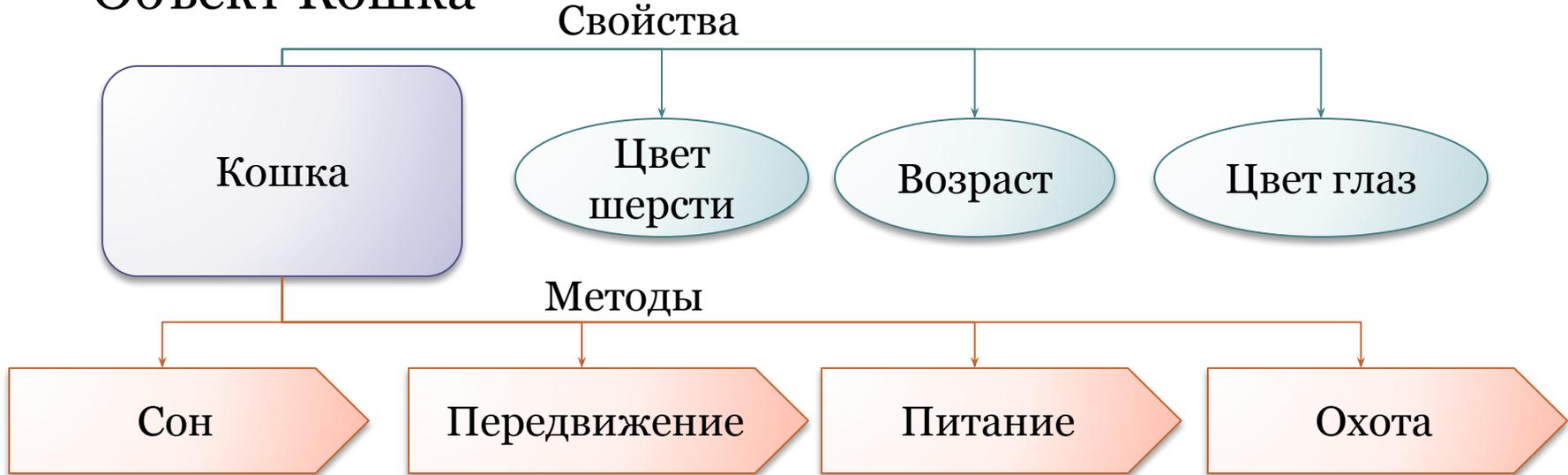
Объекты



Объекты — это еще один вид данных, является как и массив контейнером, только в себе может еще заключать ещё и функции (функции объекта обычно называют методами)

- Объекты еще больше приближают абстракцию к реальности
- С большим кодом легче работать при помощи объектов, т.к. редактировать придется не весь код, а только отдельный метод или свойство (переменные в объектах)
- Объектами проще управлять оставляя интерфейс для управления, пряча реализацию внутри них

Объект Кошка



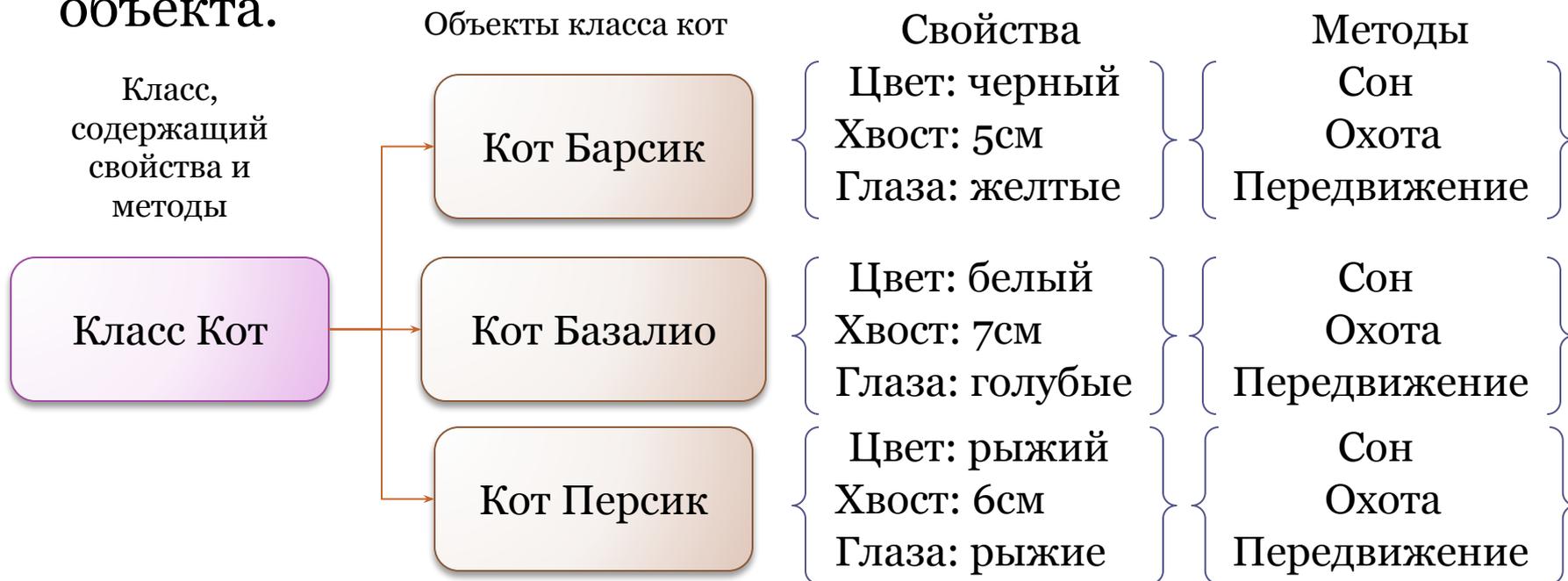
Объекты позволяют моделировать «Объекты» из реального мира. При названии свойств и методов, названия совпадают с реальными именами действий и качеств реального объекта.

Удобство при использовании объектов: скажем нам нужно изменить виртуальное передвижение нашей виртуальной кошки, тогда, не обращая на весь остальной код, мы меняем только метод «Передвижение». Чем меньше код над которым мы работаем в данный момент, тем он понятнее и тем быстрее и проще его дополнить или изменить

Создание (инициализация) объекта

Объект существует только тогда когда мы его создадим. Как и с функциями, пока их не вызовешь, они представляют из себя пока только участок кода.

Как и для массивов, для объектов существуют особые конструкции для того, чтобы их создать. Объекты создаются из классов. Класс — это начальный каркас объекта.



<?

Class Cat

```
{  
    public $name;  
    public $color;  
    public $tail;  
    public $eye;  
  
    function setParameters($name, $color, $tail, $eye)  
    {  
        $this->name = $name;  
        $this->color = $color;  
        $this->tail = $tail;  
        $this->eye = $eye;  
    }  
}
```

```
function sleep()  
{  
  
}
```

```
function movement()  
{  
  
}
```

```
function hunting()  
{  
  
}
```

}

```
$firstCat = new Cat();  
$firstCat->setParameters('Барсик', 'Черный', 5, 'желтый');  
$secondCat = new Cat();  
$secondCat->setParameters('Базилио', 'Белый', 7, 'Голубой');  
$thirdCat = new Cat();  
$thirdCat->setParameters('Персик', 'Рыжий', 6, 'Рыжий');  
  
echo '<PRE>';  
print_r($firstCat);  
print_r($secondCat);  
print_r($thirdCat);  
echo '</PRE>';  
?>
```

Cat Object

```
(  
    [name] => Барсик  
    [color] => Черный  
    [tail] => 5  
    [eye] => Желтый  
)
```

Cat Object

```
(  
    [name] => Базилио  
    [color] => Белый  
    [tail] => 7  
    [eye] => Голубой  
)
```

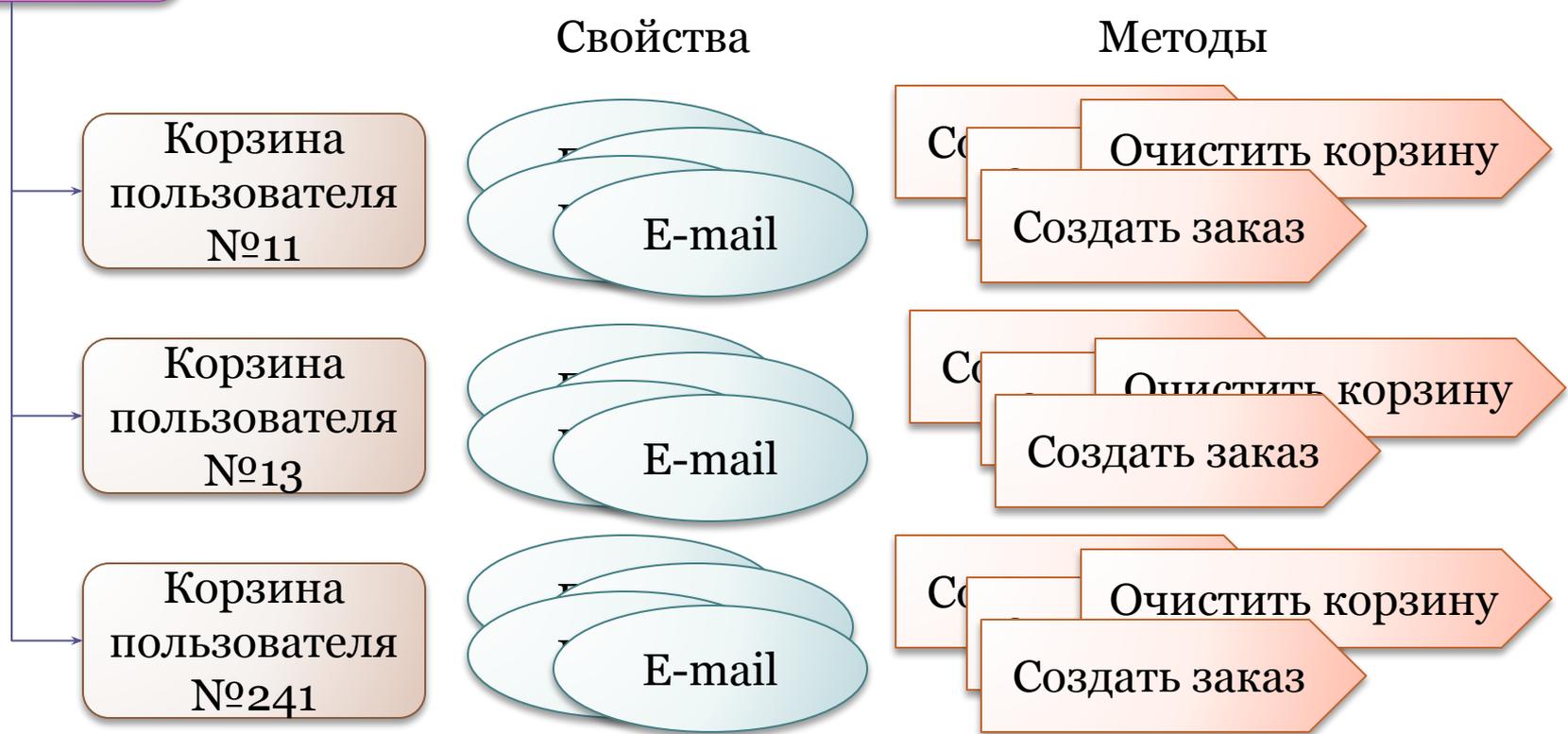
Cat Object

```
(  
    [name] => Персик  
    [color] => Рыжий  
    [tail] => 6  
    [eye] => Рыжий  
)
```

- **Class** – ключевое слово для создания класса. Все, что находится в блоке далее (в фигурных скобках), относится к данному классу.
- **Public** – ключевое слово, говорящее, к переменной можно обращаться вне объекта
- **Private** – ключевое слово, обозначающее, что к переменной можно обращаться только внутри объекта
- **\$this** – обращение к текущему объекту
- Вывести данные об объекте также можно с помощью функций **var_dump** и **print_r**
- **->** - такой символ подразумевает, что мы обращаемся к свойству объекта (переменной) или к методу данного объекта (функции)

Класс
Корзина

Пример использования объектов в интернет-магазине



Используемые классы в интернет-магазине

Класс
Корзина

Класс
Пользователь

Класс
Админка

Класс
Генерация URL

Класс
Ресайз картинок

Класс
Отправка почты

Класс
Товар

Класс
Категория товара

Класс
Экспорт товара

Работа с файлами



Работа с файлами бывает полезна для:

- Ведения счетчиков скачиваний
- Ведения логов

Файлы могут заменять базу данных, а потому в них можно хранить:

- Статьи
- Шаблоны верстки
- Экспортируемые данные с других сайтов, например котировки валют и другое

Работа с файлами разделяется на 3 этапа:

1. Открытие файла.
2. Манипуляции с данными.
3. Заккрытие файла.

Для того чтобы открыть файл используется функция **fopen()**. Обязательными параметрами этой функции является имя файла и режим файла.

```
$fp = fopen('counter.txt', 'r');
```

Согласно документации РНР выделяют следующие виды режимов файлов:

- r – открытие файла только для чтения.
- r+ - открытие файла одновременно на чтение и запись.
- w – создание нового пустого файла. Если на момент вызова уже существует такой файл, то он уничтожается.
- w+ - аналогичен r+, только если на момент вызова файл такой существует, его содержимое удаляется.
- a – открывает существующий файл в режиме записи, при этом указатель сдвигается на последний байт файла (на конец файла).
- a+ - открывает файл в режиме чтения и записи при этом указатель сдвигается на последний байт файла (на конец файла). Содержимое файла не удаляется.

Записывать данные в файл при помощи PHP можно при помощи функции **fwrite()**. В качестве обязательных параметров выступает дескриптор файла и режим файла:

```
<?php
$fp = fopen("counter.txt", "a"); // Открываем файл в режиме записи
$mytext = "Это строку необходимо нам записать\r\n"; // Исходная строка
$test = fwrite($fp, $mytext); // Запись в файл
if($test)
    echo 'Данные в файл успешно занесены.';
else
    echo 'Ошибка при записи в файл.';
fclose($fp); //Закрытие файла
?>
```

Для построчного считывания файла используют функцию [fgets\(\)](#).

```
<?php
$fp = fopen("counter.txt", "r"); // Открываем файл в режиме чтения
if($fp)
{
    while (!feof($fp))
    {
        $mytext = fgets($fp, 999);
        echo $mytext."<br />";
    }
}
else
    echo "Ошибка при открытии файла";
fclose($fp);
?>
```

Заккрытие файла происходит с помощью функции [fclose\(\)](#)

```
<?php
$fp = fopen("counter.txt", "r");
if($fp)
{
    echo 'файл открыт';
    fclose($fp); // закрытие файла
}
?>
```