PHP 5

Object Model

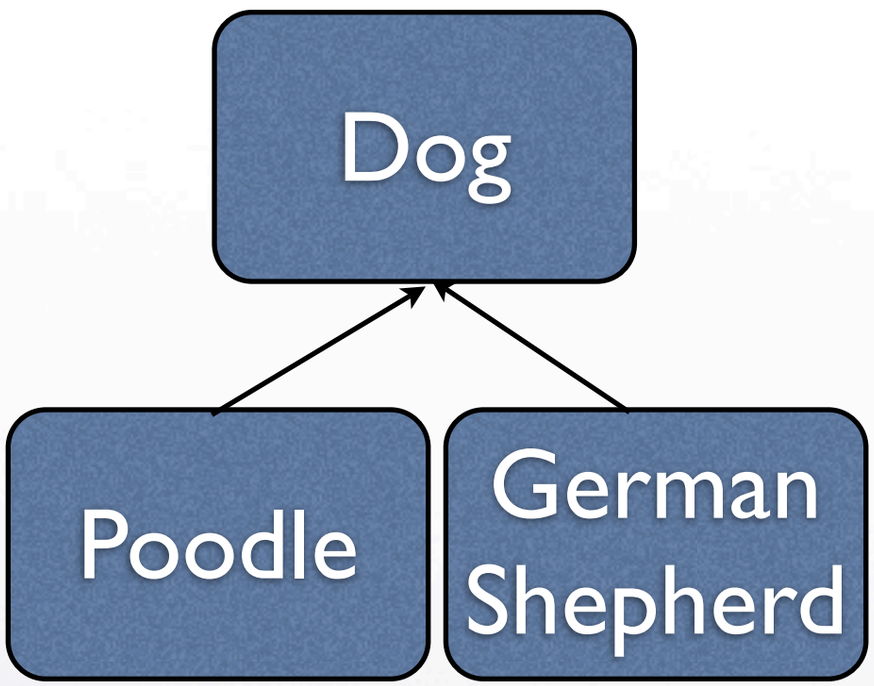
PHP 4

- Focus on procedures
  - Object orientation only rudimentary
- Usually only one file for all the code
- Big effort for refactoring necessary

Object Orientation

- Abstraction from real world objects
  - Classes define object layouts, behavior
    - Properties and methods
  - Objects are instances of said classes
    - Contain data and state information
    - Understand messages sent to them

Real World Example



Observations:

- All dogs can bark
- There is nothing like a "Dog" as such
  - abstract class?

PHP 5

- Object orientation was reintroduced, improved

```
class dog {
  public $name = "a dog";

  public function bark() {
    echo "Woof Woof";
  }

}
```

keyword **class**

property

keyword **function**

Inheritance

- PHP5 supports single inheritance

```
class Poodle extends Dog {
}
```

keyword **extends**

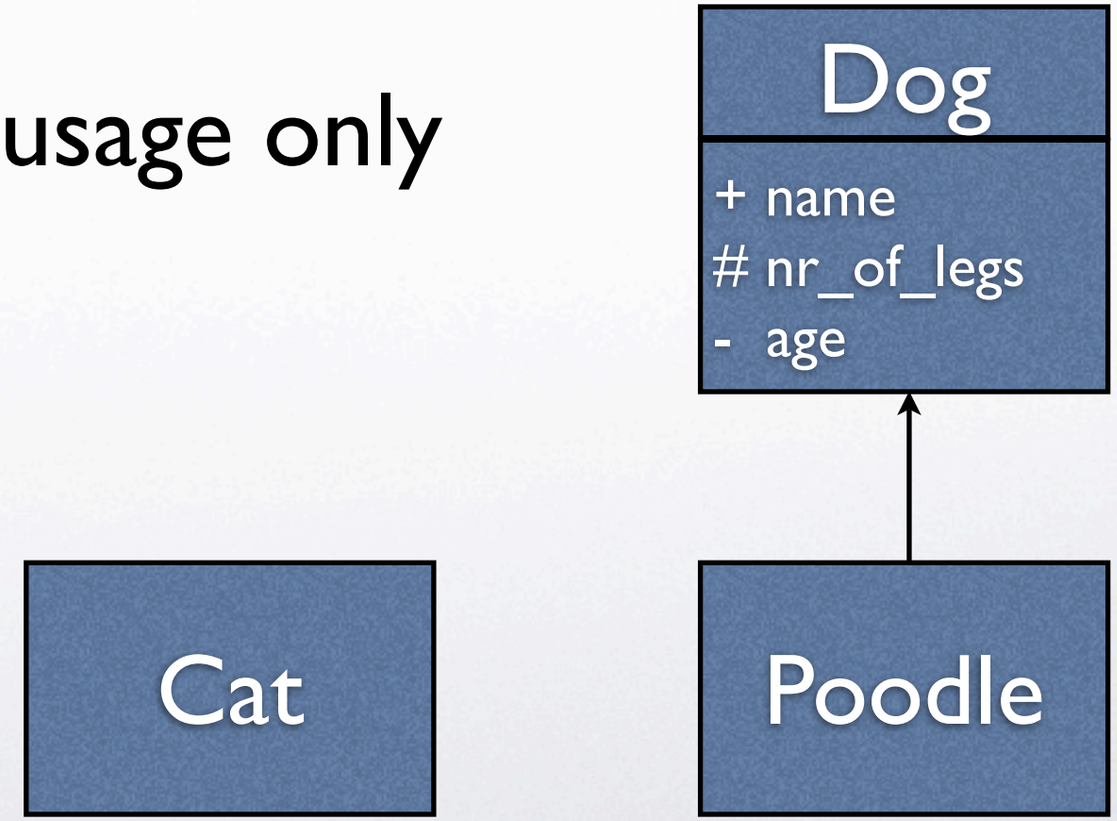
A poodle *inherits* the property name and the method bark().

```
$myPet = new Poodle();
$myPet->bark();
```

Woof woof

Visibilities

- + public > reachable from everywhere
- # protected > reachable from inherited classes
- - private > internal usage only



Overloading

- Poodles are dogs, all dogs bark, but poodles bark different!

```
class Poodle extends Dog {
  public function bark() {
    echo "Wiif wiif";
  }
}

$myPet = new Poodle();
$myPet->bark();
```

function was overloaded

Wiif wiif

Overloading II

- Poodles are dogs, and bark similar to other dogs

```
class Poodle extends Dog {
  public function bark() {
      parent::bark();
      echo "Wiif wiif";
  }
}
$myPet = new Poodle();
$myPet->bark();
```

Function from parent class Dog is still called

Woof woof
Wiif wiif

Creation & Removal

- Constructors __construct()
  - Initialize instances, e.g. by establishing DB connections before usage
- Deconstructors __destruct()
  - Preperations before removing an object, e.g. closing remaining DB connections

Using a Constructor

- All poodles need a name

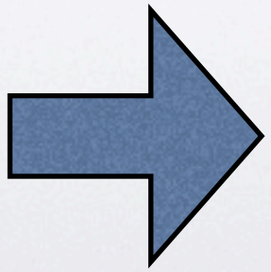
```
class Poodle extends Dog {
  public $name = "";
  public function __construct($param) {
    $this->name = $param;
  }
}
$myPet = new Poodle("Fluffy");
$myPet->sayName();
```

File Management

- One file for each class
  - Which to use? where to find them?
- Includes for each needed file -> long list

```
function __autoload($class_name) {
  require_once $class_name. '.php';
}

$c = new MyClass();
```



MyClass.php is included automatically

Thanks for Your Attention

Any questions left?