



PHP Level 2

Урок 2

ООП в PHP.

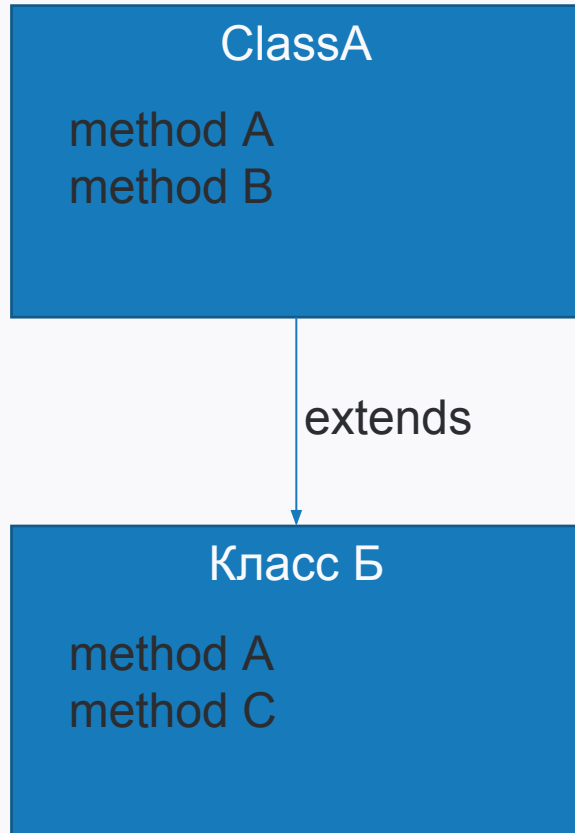
Расширенное изучение.

План урока

1. Ключевые слова
2. Абстрактные классы и интерфейсы
3. Перегрузка и магические методы
4. Контроль типов и пространства имён
5. Трейты
6. Паттерн Singleton



Ключевое слово parent



```
<?php
class BaseClass {
    function __construct() {
        echo "Конструктор класса BaseClass\n";
    }
}

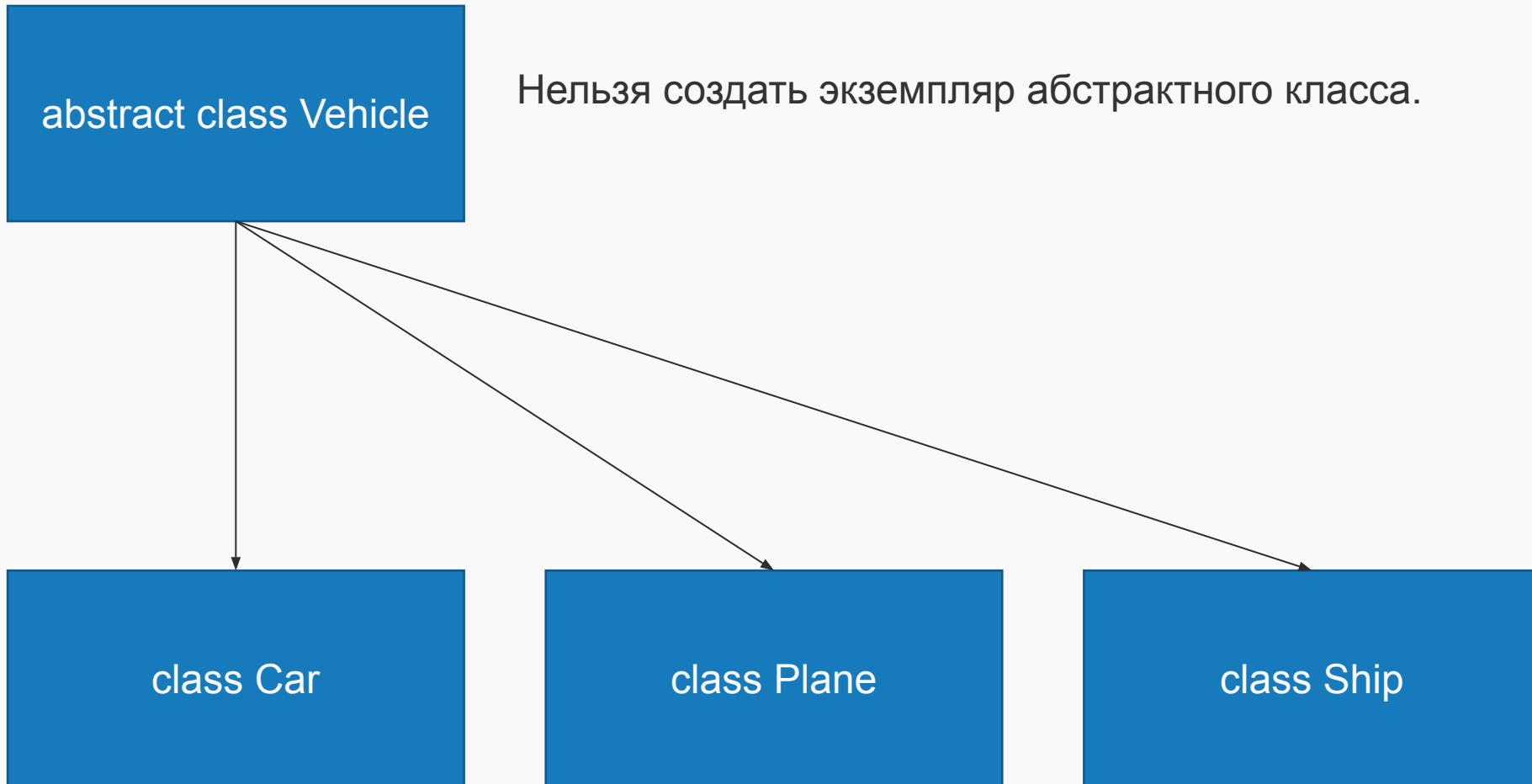
class SubClass extends BaseClass {
    function __construct() {
        parent::__construct();
        echo "Конструктор класса SubClass\n";
    }
}

$obj = new BaseClass();
$obj = new SubClass();

?>
```



Абстрактные классы



Интерфейсы

interface CarTemplate

С помощью интерфейсов можно **описать** методы, которые должны быть реализованы в классе без необходимости описания их функционала.

class Audi



Перегрузка и магические методы

Перегрузка в PHP дает возможность динамически "создавать" свойства и методы. Такие методы и свойства обрабатываются с помощью "волшебных" методов, которые можно создать в классе для различных видов действий.

```
<?php  
  
public void __set (string $name , mixed $value)  
  
public mixed __get (string $name)  
  
?>
```



Контроль типа

Тип	Описание	Минимальная версия PHP
Имя класса/интерфейса	Аргумент должен быть instanceof , что и имя класса или интерфейса.	PHP 5.0.0
<i>self</i>	Этот параметр должен быть instanceof того же класса, в методе которого он указан. <i>self</i> можно использовать только в функциях класса либо объекта класса.	PHP 5.0.0
array	Аргумент должен быть типа array .	PHP 5.1.0
callable	Аргумент должен быть корректным callable типом.	PHP 5.4.0
bool	Аргумент должен быть типа boolean .	PHP 7.0.0
float	Аргумент должен быть float типа.	PHP 7.0.0
int	Аргумент должен быть типа integer .	PHP 7.0.0
string	Аргумент должен иметь тип string .	PHP 7.0.0

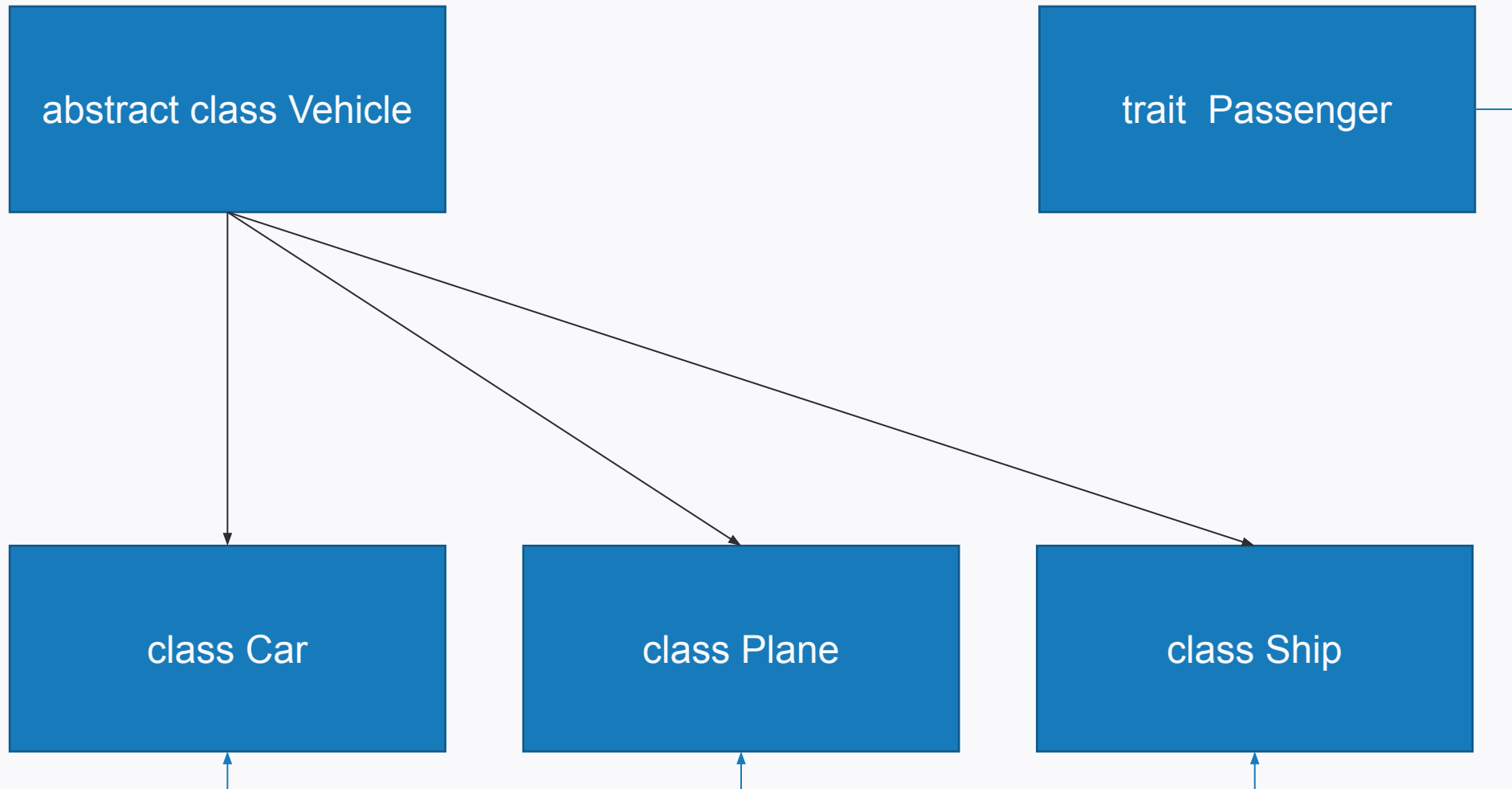


Пространства имён

Пространства имен, это один из способов инкапсуляции элементов.



Трейты



Singleton

Шаблон проектирования или **паттерн** в разработке программного обеспечения — повторяемая архитектурная конструкция, представляющая собой решение проблемы проектирования в рамках некоторого часто возникающего контекста.

Одиночка (англ. **Singleton**) — шаблон проектирования, гарантирующий, что в однопоточном приложении будет единственный экземпляр класса с глобальной точкой доступа.

