

3. Дополнительные механизмы ООП

Константы класса

Статические свойства и методы

Определяются в контексте класса, а не объекта.

- Доступны из любой точки программы.
- Доступны каждому экземпляру данного класса.
- Не нужно создавать экземпляры класса только ради вызова простой функции.

Примеры реализации на PHP

Интерфейсы

В интерфейсе декларируется функциональность, которую должен реализовать каждый класс, реализующий этот интерфейс.



Что здесь?



Что здесь?





Что здесь?

Любая реализация интерфейса
«Вилка для электрической розетки»

Интерфейсы

Примеры на PHP

Интерфейсы

Аналог абстрактного класса, содержащего только абстрактные методы. Интерфейсы — это облегченные классы, которые диктуют нам условия.

Интерфейсы можно наследовать друг от друга.

Класс может реализовывать несколько интерфейсов (наследоваться от нескольких классов нельзя).

Класс = тип данных + шаблон для создания экземпляров
Интерфейс = тип данных

Трейты (примеси)

Трейт — общий программный код, который можно добавлять к любым классам.

Аналог `include`, действие которого распространяется только на конкретный класс.

Трейты не изменяют тип класса, в который они включаются.

Вместо трейтов код можно вынести в статические методы какого-то класса и вызывать эти методы в других классах.

Интерфейсы и трейты в реальной жизни

Трейты — повторное использование кода.

Интерфейсы — расширяемость решений.

Можно написать библиотеку с функциональностью для определенного интерфейса, и использовать ее для любых реализаций этого интерфейса.

Пример: `vendor/laravel/framework/src/Illuminate/Database/Schema`

Пространства имен для классов

Возможность группировать логически связанные классы, интерфейсы, функции и константы.

Для решения двух проблем:

- Конфликт имен между вашим кодом и внутренними классами/функциями/константами PHP или сторонними.
- Возможность создавать псевдонимы для длинных имен.