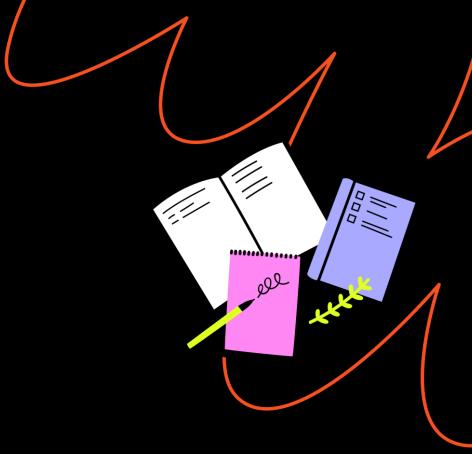
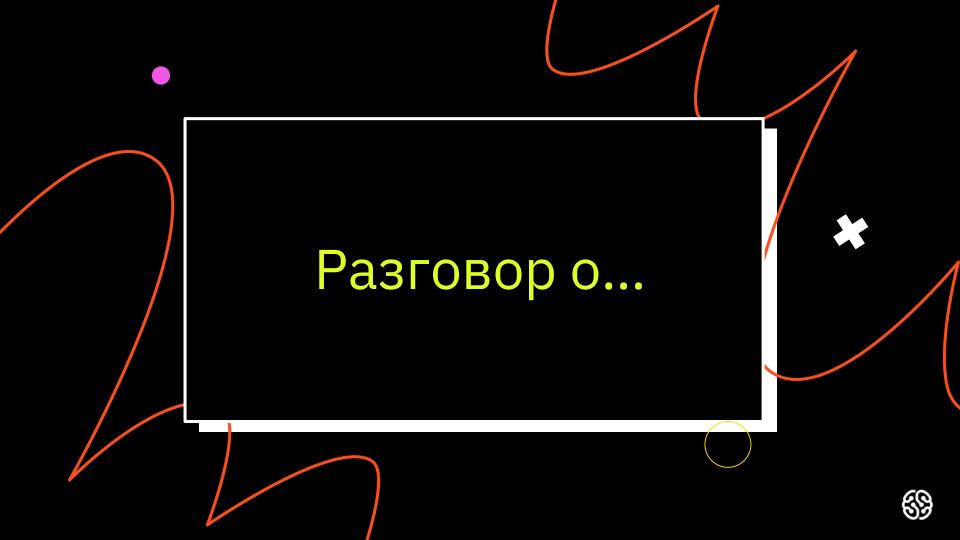


ООП: Абстракция и интерфейсы



Важный принцип



Разговор о...

- 1. Введение
- 2. Абстракция
- 3. Абстрактный класс
- 4. Абстрактный метод
- 5. Интерфейс
- 6. Примеры
- **7**. Итоги





Введение

- Инкапсуляция
- Наследование
- Полиморфизм



Введение

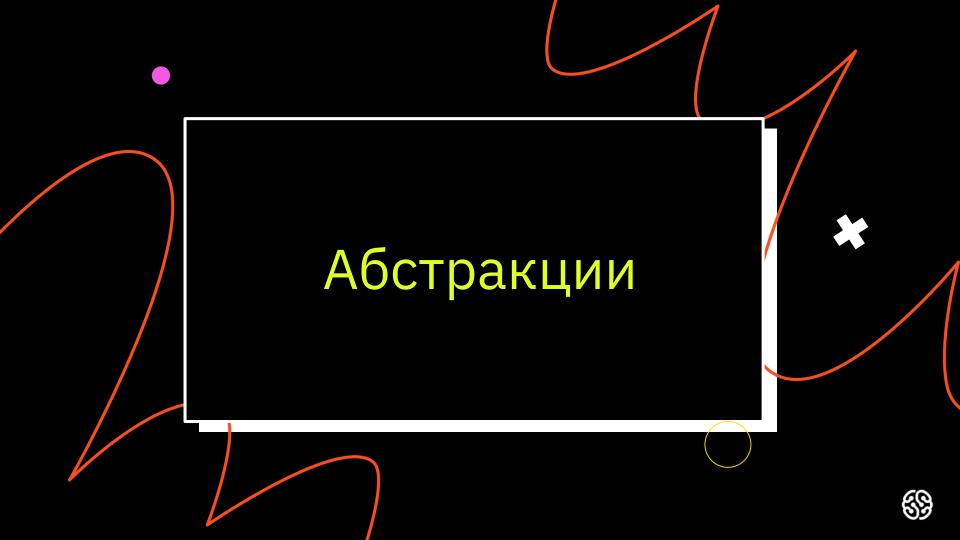
- Инкапсуляция
- Наследование
- Полиморфизм
- Абстракция



Введение

- Инкапсуляция
- Наследование
- Полиморфизм
- Абстракция
- * переиспользование кода
- * отправка сообщений





Абстракция — это свойство системы, позволяющее описать общие характеристики базового класса для всех его производных классов, а наполнение деталями предоставляется каждому из этих классов.



Абстракция — это свойство системы, позволяющее описать общие характеристики базового класса для всех его производных классов, а наполнение деталями предоставляется каждому из этих классов.

В абстрактном классе определяются лишь общие поля и характер методов, которые должны быть конкретно реализованы в производных классах, а не в самом базовом классе.



Абстракция — это свойство системы, позволяющее описать общие характеристики базового класса для всех его производных классов, а наполнение деталями предоставляется каждому из этих классов.

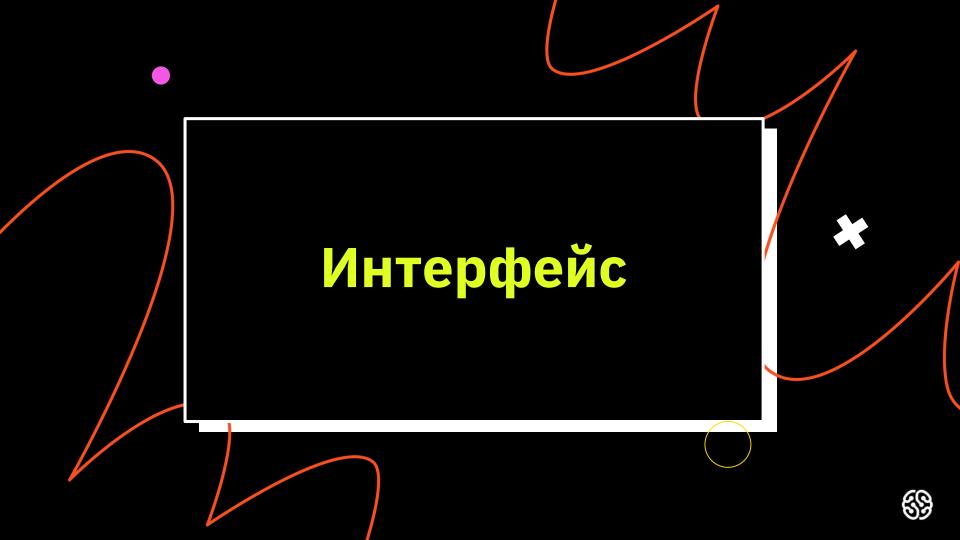
В абстрактном классе определяются лишь общие поля и характер методов, которые должны быть конкретно реализованы в производных классах, а не в самом базовом классе.

```
public abstract class Model {
    public abstract void Method();
}
```



Демонстрация





interface предназначен для описания исключительно общего поведения сущностей.



interface предназначен для описания исключительно общего поведения сущностей.

Частный случай: для гарантии, реализации некоторого поведения.



interface предназначен для описания исключительно общего поведения сущностей.

Частный случай: для гарантии, реализации некоторого поведения.

Интерфейс – это частный случай класса. Он представляет собой полностью абстрактный класс с абстрактными членами.



interface предназначен для описания исключительно общего поведения сущностей.

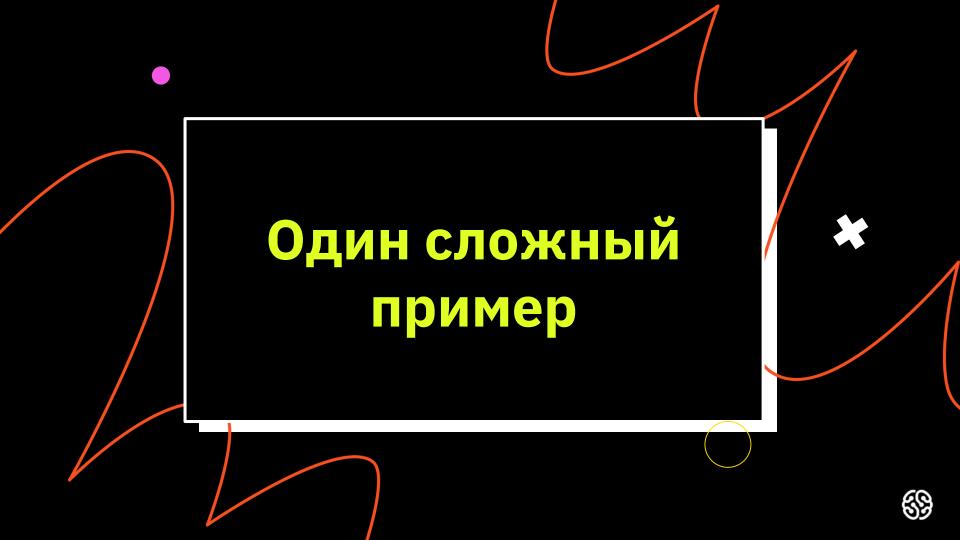
Частный случай: для гарантии, реализации некоторого поведения.

- * Аналогичен абстрактному классу со всеми абстрактными методами, но допускается множественное наследование.
- * расширение поведения
- ^{*} что я могу делать



Демонстрация





Один сложный пример

Демонстрация Ех006





Класс

- 1 Нужно больше времени уделять объектному мышлению.
- **2** Нужно больше писать код.
- 3 Нужно больше времени уделять объектному мышлению и написанию кода.



Спасибо // за внимание /

