(/me)

Лекции

Карта квестов (/quests)

Список лекций (/quests/lectures)

CS50 (/quests/QUEST HARVARD CS50)

Android (/quests/QUEST\_GOOGLE\_ANDROID)

C

## Как пользоваться абстрактными классами

Java Core (/quests/QUEST\_JAVA\_CORE)
3 уровень (/quests/lectures/?quest=QUEST\_JAVA\_CORE&level=3), 3 лекция (/quests/lectures/questcore.level03.lecture03)

ОТКРЫТА

— Привет, Амиго! Вчера ты уже слушал лекцию про абстрактные классы. Теперь пришло время углубить наши познания. Хочу научить тебя правильно пользоваться абстрактными классами.

Сложно представить аналогию абстрактного класса в реальной жизни. Обычно класс является моделью какой-нибудь сущности. Но абстрактный класс содержит не только реализованные методы, но и не реализованные. Что же это значит? Аналогом чего является абстрактный класс и есть ли у него аналоги в реальном мире?

На самом деле есть. Представь себе почти законченный кузов машины на конвейере. Туда могу поставить как спортивный двигатель, так и экономичный. Как кожаный салон, так и матерчатый. Конкретная реализация машины еще не определена. Более того, таких конкретных реализаций на основе этого кузова предполагается несколько. Но в таком виде машина никому не нужна. Это — классический абстрактный класс: его объекты не имеют смысла, поэтому их создание запрещено, класс имеет смысл, но только для его многочисленных полноценных наследников, которые будут созданы на его основе.

— Это не сложно.

— Но могут быть и более абстрактные аналогии. Больше похожие на интерфейсы, с несколькими реализованными методами. Например, профессия переводчик. Без уточнения, с какого, и на какой язык, получим «абстрактного переводчика в вакууме». Или телохранитель. Про него может быть известно, что он владеет восточными единоборствами и может защитить клиента. Но какими именно единоборствами, и каким способом защитить клиента — это уже «особенности реализации» каждого конкретного телохранителя.

Давай посмотрим пример:

Код на Java

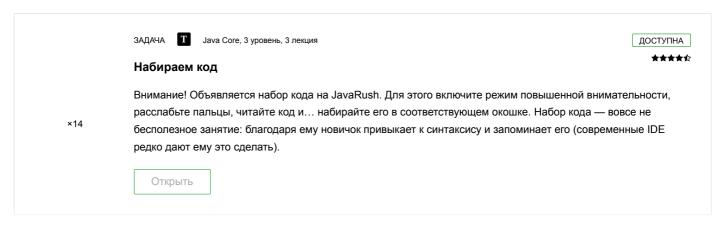
Описание

```
1 abstract class BodyGuard
2 {
    abstract void applyMartialArts(Attacker attacker);
3
4
5
    void shoot(Attacker attacker)
6
    {
7
       gun.shoot(attacker);
8
    }
9
10
    void saveClientLife(Attacker attacker)
11
     if (attacker.hasGun())
12
13
        shoot(attacker);
14
     else
15
        applyMartialArts(attacker);
16
    }
17 }
```

В классе «телохранитель» определено, как поступать в случае нападения: стрелять или применить восточные единоборства.

Но не определено, какие именно восточные единоборства, хотя точно известно, что этот навык есть.

Мы можем создать несколько разных телохранителей (унаследовав этот класс). Все они будут уметь защищать клиента и стрелять в нападающего.



— Действительно, по смыслу очень напоминает интерфейс с несколькими реализованными методами.

— Да, абстрактные классы такого типа мы будем часто встречать среди стандартных классов JavaSE.

< (/quests/lectures/questcore.level03.lecture02)

 $\times 13 \rightarrow (/quests/lectures/questcore.level03.lecture04)$ 



<u>ush.ru/)</u> G+ (https://plus.google.com/114772402300089087607) 

✓ (https://twitter.com/javarush\_ru) (https://twitter.com/javarush\_ru) (https://twitter.com/javarush\_ru)



Программистами не рождаются © 2018