**Как пользоваться абстрактными классами**

[Java Core](https://javarush.ru/quests/QUEST_JAVA_CORE)

[Уровень 3](https://javarush.ru/quests/lectures?quest=QUEST_JAVA_CORE&level=3), Лекция 3

— 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 могут 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 боевыми искусствами 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** |
| abstract class BodyGuard  {  abstract void applyMartialArts(Attacker attacker);  void shoot(Attacker attacker)  {  gun.shoot(attacker);  }  void saveClientLife(Attacker attacker)  {  if (attacker.hasGun())  shoot(attacker);  else  applyMartialArts(attacker);  }  } | В классе «телохранитель» определено, как поступать в случае нападения: стрелять или применить боевые искусства.  Но не определено, какие именно боевые искусства, 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. |

3

Задача

Java Core,  3 уровень,  3 лекция

Набираем код Ӏ Java Core: 3 уровень, 3 лекция

Java Core: 3 уровень, 3 лекция. Внимание! Объявляется набор кода на JavaRush. Для этого включите режим повышенной внимательности, расслабьте пальцы, читайте код и… набирайте его в соответствующем окошке. Набор кода — вовсе не бесполезное занятие: благодаря ему новичок привыкает к синтаксису и запоминает его (современные IDE редко дают ему это сделать).

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

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