



Абстрактные классы

Java Core (/quests/QUEST_JAVA_CORE)
2 уровень (/quests/lectures/?quest=QUEST_JAVA_CORE&level=2), 5 лекция (/quests/lectures/questcore.level02.lecture05)

ОТКРЫТА

— Привет, Амиго! Новая интересная тема.

— Да сегодня просто день интересных тем!!

— Спасибо!

— Да не за что.

— Помнишь ситуацию, когда мы ввели базовый класс `ChessItem` для упрощения всех классов шахматных фигур?

— Да.

— Теперь представь, что у каждой фигуры есть метод, который занимается ее отрисовкой на экране. Вызываешь метод, и фигура сама себя рисует в своих текущих координатах. Удобно было бы вынести этот метод в базовый класс?

— Да. После того, что я узнал о полиморфизме, можно было бы вызывать метод отрисовки для всех фигур, независимо от их типа. Примерно так:

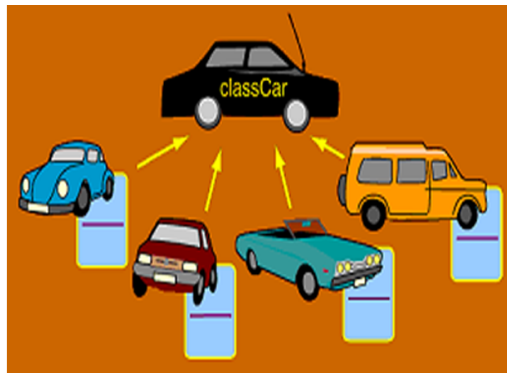
Пример

```
1 class ChessBoard
2 {
3     public void drawAllChessItems()
4     {
5         //добавляем фигуры в список
6         ArrayList items = new ArrayList();
7         items.add(new King());
8         items.add(new Queen());
9         items.add(new Bishop());
10
11         //рисуем их независимо от их типа.
12         for (ChessItem item: items)
13         {
14             item.draw();
15         }
16     }
17 }
```

— Молодец. Именно так. А что бы отрисовал на экране метод `draw` самого класса `ChessItem`?

— Не знаю. Такой фигуры ведь в шахматах нет. Значит, и изображения у нее нет.

— Именно. Более того, создавать объекты типа `ChessItem` — не имеет смысла. Это не фигура из шахмат, всего лишь абстракция — класс, который мы сделали для удобства. Так работает абстракция из ООП: мы вынесли важные (общие для всех фигур) данные и методы в базовый класс, а их различия оставили в их классах.



Для такого случая в Java есть специальный тип классов – абстрактные классы. Вот три вещи, которые стоит помнить об абстрактных классах.

1) Абстрактный класс может содержать объявление метода без его реализации. Такой метод называется абстрактным.

Пример

```

1  public abstract class ChessItem
2  {
3      public int x, y; //координаты
4      private int value; // «ценность» фигуры
5
6      public int getValue() //обычный метод, возвращает значение value
7      {
8          return value;
9      }
10
11     public abstract void draw(); //абстрактный метод. Реализация отсутствует.
12
13 }
```

ЗАДАЧА **T** Java Core, 2 уровень, 5 лекция

ДОСТУПНА

★★★★☆

Набираем код

×14

Иногда думать не надо, строчить надо! Как ни парадоксально звучит, порой пальцы «запоминают» лучше, чем сознание. Вот почему во время обучения в секретном центре JavaRush вы иногда встречаете задания на набор кода. Набирая код, вы привыкаете к синтаксису и зарабатываете немного материи. А ещё — боретесь с ленью.

Открыть

2) Абстрактный метод помечается специальным ключевым словом `abstract`.

Если в классе есть хоть один абстрактный метод, класс тоже помечается ключевым словом `abstract`.

3) Создавать объекты абстрактного класса нельзя. Такой код просто не скомпилирует.

Код на Java

Описание

```

1  ChessItem item = new ChessItem();
2  item.draw();
```

Этот код не скомпилируется

```
1 ChessItem item = new Queen();  
2 item.draw();
```

А так можно.

4) Если ты наследовал свой класс от абстрактного класса, то нужно переопределить все унаследованные абстрактные методы — написать для них реализацию. Иначе такой класс тоже придется объявить абстрактным. Если в классе есть хотя-бы один нереализованный метод, объявленный прямо в нем или унаследованный от класса-родителя, то класс считается абстрактным.

— А зачем это все нужно? Зачем нужны абстрактные классы? Разве нельзя вместо них использовать обычные? А вместо абстрактных методов просто писать две скобочки в качестве тела метода — {} ?

— Можно. Но эти ограничения сродни модификатору `private`. Мы специально запрещали с помощью `private` прямой доступ к данным, чтобы другие программисты и их классы пользовались только написанными нами `public`-методами.

То же и с абстрактным классом. Тот, кто написал этот класс, не хочет, чтобы создавались его объекты. Наоборот, он рассчитывает на то, чтобы от его абстрактного класса наследовались и переопределяли абстрактные методы.

— Все равно не понятно, зачем усложнять себе жизнь?

— Преимущество этого проявляется в больших проектах. Чем больше классов, тем четче приходится очерчивать их роли. Ты увидишь преимущество этого, и уже в ближайшем будущем. Все через это проходят.

[< \(/quests/lectures/questcore.level02.lecture04\)](/quests/lectures/questcore.level02.lecture04)

[×12 > \(/quests/lectures/questcore.level02.lecture06\)](/quests/lectures/questcore.level02.lecture06)

старые

25 марта, 22:56

Здесь подробно и ясно написано про абстрактные методы, а заодно и про интерфейсы:
<http://kostin.ws/java/java-abstract-and-interfaces.html>

0

7 сентября 2017, 17:19

- 1)
- 2)
- 3)
- 4)

)))))) прям как в армии!

+65

11 сентября 2017, 17:28

Улыбнуло!)

0

11 сентября 2017, 23:07

- 0)
- 1)
- 2)
- 3)

+25

12 сентября 2017, 15:35

Мы же считаем вещи по их числу, а не последнему индексу? :)

+1

15 сентября 2017, 23:28

Это вам не баги искать в коде:) тут мозги нужны чтобы заметить, что вместо 3х пунктов представлены 4:)

+2

24 октября 2017, 17:06

```
"0)  
1)  
2)  
3)"  
твойМассив.length == 4 ;)
```

+2

5 декабря 2017, 04:48

Мне сразу бросилось в глаза. Подумал что за бред!
А еще как принято в программировании считать с нуля, даже если так то всеравно 4 выйдет дойдя до 3.
Кароче забил голову сам себе, из-за какого-то парнишки =)

0

22 февраля, 16:47

[illegible]

0

3 июня 2017, 11:07

Т.е если я правильно понимаю,то все классы унаследованные от ChessItem тоже становятся абстрактными?И при этом мы должны переопределить все методы?

Почему бы просто не объявить класс private ?

Ответить

0

Даня Кельвич 39 уровень, Киев

4 июня 2017, 10:22

Нет, классы унаследованные от ChessItem реализуют все абстрактные методы, поэтому объекты этих классов можно создать. Это все объясняется в Head First Java.

Ответить

+2

Akira Rokudo 20 уровень, Москва

25 июня 2017, 08:07

Добавлю к комментарию Дани-либо реализуют все методы, либо тоже становятся абстрактными. Почему же не объявлять класс как private? это возможно только если он внутренний. (Иначе он будет доступен только для себя, что лишено логики) А смысл внутренних классов в моем понимании несколько другой, чем шаблонный класс от которого наследуются реализуемые шаблоны)

Ответить

+2

Katsiaryna Bazulka 35 уровень

23 мая 2017, 20:53

Задача на написание кода не работает. В окне "Код, который надо повторить:" ничего нет, а в "Ваш код:" ничего нельзя ввести.

Ответить

0

Vikentsi 22 уровень, Минск

20 июля 2017, 19:46

все работает

Ответить

+1

Евгений 32 уровень

25 октября 2017, 17:01

у меня такая же проблема, но только в браузере IE. в других все в порядке

Ответить

0

Сергей Буглак 19 уровень

23 апреля 2017, 08:38

Интересно, аж тапки горят)

Ответить

0

javarush.ru/, [G+ \(https://plus.google.com/114772402300089087607/\)](https://plus.google.com/114772402300089087607/), [Twitter \(https://twitter.com/javarush_ru\)](https://twitter.com/javarush_ru), [Facebook \(https://www.facebook.com/javarush\)](https://www.facebook.com/javarush)



Программистами не рождаются
© 2018