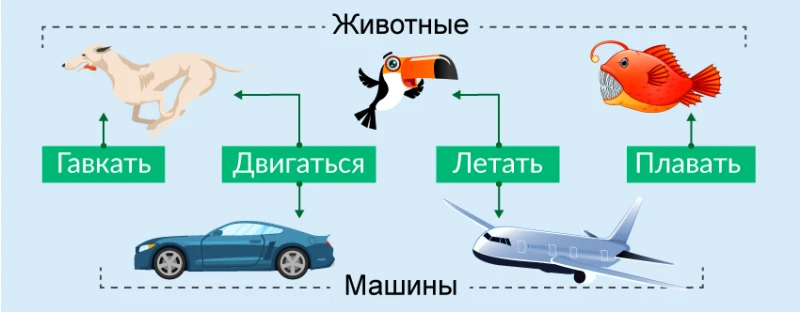
**Интерфейсы — это больше чем интерфейсы — это поведение**

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

[Уровень 2](https://javarush.ru/quests/lectures?quest=QUEST_JAVA_CORE&level=2), Лекция 8

— Привет, Амиго! А вот и снова я. Хочу рассказать тебе еще об одном взгляде на интерфейсы. Понимаешь, класс – это, чаще всего модель какого-то конкретного объекта. Интерфейс же больше соответствует не объектам, а их способностям или ролям.

[](https://cdn.javarush.ru/images/article/da234191-c274-4136-99da-eb0c438d0683/original.jpeg)

Например, такие вещи, как машина, велосипед, мотоцикл и колесо лучше всего представить в виде классов и объектов. А такие их способности как «могу ездить», «могу перевозить людей», «могу стоять» — лучше представить в виде интерфейсов. Смотри пример:

|  |  |
| --- | --- |
| **Код на Java** | **Описание** |
| **interface** Moveable  {  void move(String newAddress);  } | — соответствует способности передвигаться. |
| **interface** Driveable  {  void drive(Driver driver);  } | — соответствует способности управляться водителем. |
| **interface** Transport  {  void addStuff(Object stuff);  Object removeStuff();  } | — соответствует способности перевозить грузы. |
| class Wheel **implements** Moveable  {  ...  } | — класс «колесо». Обладает способностью передвигаться. |
| class Car **implements** Moveable, Drivable, Transport  {  ...  } | — класс «машина». Обладает способностью передвигаться, управляться человеком и перевозить грузы. |
| class Skateboard **implements** Moveable, Driveable  {  ...  } | — класс «скейтборд». Обладает способностью передвигаться и управляться человеком. |

3

Задача

Java Core,  2 уровень,  8 лекция

Набираем код Ӏ Java Core: 2 уровень, 8 лекция

Java Core: 2 уровень, 8 лекция. 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.

Интерфейсы сильно упрощают жизнь программиста. Очень часто в программе тысячи объектов, сотни классов и всего пара десятков интерфейсов – ролей. Ролей мало, а их комбинаций – классов – очень много.

Весь смысл в том, что тебе не нужно писать код для взаимодействия со всеми классами. Тебе достаточно взаимодействовать с их ролями (интерфейсами).

Представь, что ты – робот-строитель и у тебя в подчинении есть десятки роботов, каждый из которых может иметь несколько профессий. Тебе нужно срочно достроить стену. Ты просто берешь всех роботов, у которых есть способность «строитель» и говоришь им строить стену. Тебе все равно, что это за роботы. Хоть робот-поливалка. Если он умеет строить – пусть идет строить.

Вот как это выглядело бы в коде:

|  |  |
| --- | --- |
| **Код на Java** | **Описание** |
| **static interface WallBuilder**  {  void buildWall();  } | — способность «строитель стен». Понимает команду «(по)строить стену» — имеет соответствующий метод. |
| **static class** РабочийРобот implements WallBuilder  {  void buildWall()  {  …  }  }  **static class** РоботСторож implements WallBuilder  {  void buildWall()  {  …  }  }  **static class** Поливалка  {  …  } | — роботы у которых есть эта профессия/особенность.  — для удобства я сделал классам имена на русском. Такое допускается в java, но крайне нежелательно.  — поливалка не обладает способностью строить стены (не реализует интерфейс WallBuilder). |
| **public static void** main(String[] args)  {  //добавляем всех роботов в список  ArrayList robots = new ArrayList();  robots.add(new РабочийРобот());  robots.add(new РоботСторож());  robots.add(new Поливалка());  //строить стену, если есть такая способность  for (Object robot: robots)  {  if (robot instanceof WallBuilder)  {  WallBuilder builder = (WallBuilder) robot;  builder.buildWall();  }  }  }  } | — как дать им команду – построить стену? |

— Чертовски интересно. Даже и не думал, что интерфейсы – такая интересная тема.

— А то! В совокупности с полиморфизмом – это вообще бомба.