**Перегрузка методов**

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

[Уровень 2](https://javarush.ru/quests/lectures?quest=QUEST_JAVA_CORE&level=2), Лекция 3

— Привет, Амиго! А я расскажу тебе о перегрузке методов.

— Их еще и перегружать можно?! Да что за день сегодня такой!

— С ними много чего можно делать, но не будем об этом.

— Согласен.

— Перегрузка – это очень простая операция. На самом деле – это даже не операция над методами, хотя иногда ее называют страшным словом – **параметрический полиморфизм.**

Дело в том, что все методы внутри класса должны иметь уникальные имена.

— Да, я знаю об этом.

— Так вот, это не совсем так. Вернее совсем не так. Метод не должен иметь уникальное имя. Уникальным должно быть объединение из имени и типов параметров этого метода. Их еще называют сигнатурами методов.

|  |  |
| --- | --- |
| **Код** | **Описание** |
| public void print();  public void print2(); | Так можно. Два метода имеют уникальные имена. |
| public void print();  public void print(int n); | И так можно. Два метода имеют уникальные имена (сигнатуры). |
| public void print(int n, int n2);  public void print(int n); | Все еще уникальные методы. |
| public int print(int a);  public void print(int n); | А так нельзя. Методы не уникальные, хоть и возвращают разные типы. |
| public int print(int a, long b);  public long print(long b, int a); | А так – можно. Параметры методов уникальные. |

— Где-то я уже такое видел.

— Ага. Когда ты пишешь System.out.println, Intellij IDEA подсказывает тебе и выдает в подсказке пару десятков методов print с разными параметрами. Компилятор просто определит нужный метод, по типам переменных, которые ты туда передаешь, и пропишет вызов именно его.

— Это, вроде, не сложно. Не полиморфизм, однако.

— Точнее — не переопределение методов

Кстати, обращаю твое внимание, что имена параметров роли не играют – они теряются при компиляции. После компиляции о методе известно только его имя и типы параметров.