



## Перегрузка методов

Java Core (/quests/QUEST\_JAVA\_CORE)  
2 уровень (/quests/lectures/?quest=QUEST\_JAVA\_CORE&level=2), 3 лекция (/quests/lectures/questcore.level02.lecture03)

ОТКРЫТА

— 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

```
1 public void print();  
2 public void print2();
```

Так можно. Два метода имеют уникальные имена.

```
1 public void print();  
2 public void print(int n);
```

И так можно. Два метода имеют уникальные имена (сигнатуры).

```
1 public void print(int n, int n2);  
2 public void print(int n);
```

Все еще уникальные методы.

```
1 public int print(int a);  
2 public void print(int n);
```

А так нельзя. Методы не уникальные, хоть и возвращают разные типы.

```
1 public int print(int a, long b);  
2 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**.

< (/quests/lectures/questcore.level02.lecture02)

×12 > (/quests/lectures/questcore.level02.lecture04)



0



0



0



0



0

+20

## Комментарии (20)

популярные

новые

старые

Никита

Fs Jt 20 уровень, Киев

18 декабря 2017, 10:53



Кроме того, перегрузка использовалась в конструкторах. Конструктор, по сути, тоже метод.

Ответить

+6

Рустам Сафин 16 уровень, Казань

27 ноября 2017, 16:19



ага... уникальные имена, но вот почему

```
public int print(int a);
public void print(int n);
```

не уникальные и запрещенные, а

```
public void print(int n, int n2);
public void print(int n);
```

уникальные? Где в уникальность последних и в чем первые не уникальные? Или всё дело именно в парности методов? Но тогда в чем уникальность последнего примера:

```
public int print(int a, long b);
public long print(long b, int a);
```

Они оба принимают одинаковые параметры

Ответить

0

Markouski Andrei 16 уровень

8 декабря 2017, 14:30



Если все правильно понял.  
Сигнатура метода должна быть уникальной.  
Сигнатура метода — это имя метода плюс параметры (причем порядок параметров имеет значение).  
Модификаторы не входят в сигнатуру.  
Если есть сомнения, лучше читать документы oracle  
<https://docs.oracle.com/javase/tutorial/java/javaOO/methods.html>

Ответить

+4

Logikarma 13 уровень

27 января, 19:58



От порядка параметров будет зависеть, в каком порядке мы будем записывать аргументы при вызове метода.

например при вызове метода `print(int n, int n2);`  
вызов будет в коде будет такой: `print(5, 3);`  
**5** это аргумент для параметра **int n**  
**3** аргумент для **int n2**

записано так, потому что знали в каком порядке они заявлены в параметрах метода при его определении.

Чтобы разобраться в этом, достаточно представить, как ты будешь вызывать этот метод и какие аргументы будешь в него передавать.

Ответить

0

P0huber 14 уровень

11 ноября 2017, 23:43



Что-то уж в лекции не совсем понятно. В последнем примере, где "А так — можно. Параметры методов уникальные." Там вроде как уникальные типы методов `int print` и `long print`, но это не учитывается и не говорит, что методы уникальны. Также как и уникальна ПОСЛЕДОВАТЕЛЬНОСТЬ параметров каждого метода, т.е (`int a, long b`). Но ведь в каждом методе параметра всего два, причем `int, long` и там и там - разве это говорит об уникальности? Кто прояснит эту путаницу?

Ответить

0

Евгений 18 уровень, Москва

13 ноября 2017, 09:57



Возвращаемое значение не относится к сигнатуре. Решение какую функцию вызывать принимается на основании имени и входных параметров, в том числе и их порядка.

Ответить

+2

mila 39 уровень, Самара

10 сентября 2017, 16:33



почему перегрузка называется?

Ответить

0

Sergio 19 уровень

12 сентября 2017, 14:00



Это такой вот перевод слова **overloading**. В оригинале речь идет о **Method overloading**. По сути мы пишем множество методов, имеющих общее имя и идентичную задачу. Лексически приставка **over** чаще всего обозначает "над, сверх, поверх, выше" в значении избыточности. Например, **overdraft** - *перепасход*, **overhead** - *чрезмерные накладные расходы или что-то очень сильно/высоко над головой* (зависит от контекста), **overwrite** - *перезаписывать*.

Overload, соответственно, - перегрузка, перегружать. Семантически данная конструкция подчеркивает именно избыточность действия. Перегружая метод, мы создаем "избыточное" (в смысле более одного) число сущностей одного и того же метода, каждая из которых оперирует своим уникальным набором параметров.

Ответить

+28

mila 39 уровень, Самара

15 сентября 2017, 11:38

да я хорошо знаю английский и всё равно не совсем улавливаю логику такого термина)  
тем не менее спасибо за развернутый ответ!

Ответить

0

Roman Bogdanov 20 уровень, Днепр

23 сентября 2017, 10:00

ПЕРЕГРУЖАЕМ ИМЯ(идентификатор) метода.

Ответить

+2

Сергей Мельников 31 уровень, Санкт-Петербург

26 сентября 2017, 08:57

Перегрузка это термин взятый из лингвистики - когда одно слово имеет несколько разных смыслов.  
например Гудеть - издавать звук, Гудеть (жаргон) - праздновать, пировать. Как видим термин один и тот-же а смысл разный. Так и в программировании имя метода одно, а действия которые он будет выполнять отличаются. Чаще всего это применяется если объекты(классы объектов) разные.  
например объекты рожок и сантехник: рожок.гудеть=звук, а сантехник.гудеть=пировать.

Ответить

0

mila 39 уровень, Самара

27 сентября 2017, 11:21

не знала про такой термин, тогда да, теперь более понятно почему здесь выбрали его

Ответить

0

Вера 18 уровень, Москва

17 ноября 2017, 19:03

В лингвистике нет такого термина ("перезагрузка"), путаете с полиморфизмом: явление, когда один и тот же термин может обозначать разные понятия

Ответить

0

Сергей Мельников 31 уровень, Санкт-Петербург

18 ноября 2017, 12:49

перегрузка !=перезагрузка

Ответить

0

Евгений 15 уровень

23 ноября 2017, 21:59

mila, я понимаю о чем вы говорите. Я впервые когда услышал о "перегрузке", то подумал о чем-то перенагруженном, т.е. как будто выполняется какое-то действие сверх того, что может быть выполнено и что может привести к зависанию системы и т.п.)))

Ответить

0

Юрий Лещенко 17 уровень, Днепр

18 августа 2017, 10:38

Заметил что автор когда показывает и рассказывает про переопределение не упоминает очень важную деталь, хоть можно это и не делать с 7 джавы, но является хорошим тоном когда переопределяешь метод нужно указывать аннотацию @Override

Ответить

+20

Алексей Талызин 20 уровень

10 октября 2017, 00:52

@Override это переопределение, из предыдущей лекции

Ответить

+2

Михаил 15 уровень, Киев

8 августа 2017, 08:28

Гриша как всегда молодец!)))

Ответить

0

shooter\_2134 36 уровень

14 июля 2017, 18:14

прямо хорошо

Ответить

0

Кирилл Нигматуллин 32 уровень

17 апреля 2017, 21:07

все понятно. спасибо

Введите ответ

Ответить

0

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



Программистами не рождаются  
© 2018