Оглавление

[Лямбда](#_Toc46169856)

[Доступ к переменным и default методам](#_Toc46169857)

[Задачи с лямбда выражениями](#_Toc46169858)

[Ссылка на метод](#_Toc46169859)

# Лямбда

Реализация метода функционального интерфейса. Лямбда-оператор -> делит выражение на две части. Слева — параметры, справа — тело.

|  |  |
| --- | --- |
| 1.  2.  3.  4.  5.  6.  7.  8.  9. | public interface Operationable {  int calculate(int x, int y);  }  public static void main(String[] args) {  Operationable operation = (x, y) -> x + y;  int result = operation.calculate(10, 20);  System.out.println(result);  } |

∙ Для определения функционального интерфейса мы либо записываем результат лямбда-выражения в локальную переменную, либо передает аргументов в параметр метода. В обоих случаях тип переменной — искомый функциональный интерфейс. Далее мы можем использовать его несколько раз.

∙ Параметры лямбда-выражения должны соответствовать типу параметрам метода функционального интерфейса:

|  |  |
| --- | --- |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10. | operation = (int x, int y) -> x + y;  // типы можно не указывать  (x, y) -> x + y;  // при отсутствии параметров  () -> 30 + 20;  // при одном параметре  n -> n \* n; |

∙ Могут ничего не возвращать (при реализации метода void);

∙ Блочные лямбда-выражения обрамляются фигурными скобками. Внути можно использовать другие блоки, циклы, конструкции if, switch, создавать переменные и т. д. В отличии от однострочной записи, в блочных лямбда-выражениях оператор return применяется явным образом.

# Доступ к переменным и default методам

∙ к неизменяемым (effectively final — не обязательно помеченные как final) локальным переменным;

∙ к полям класса;

∙ к статическим переменным.

К default методам реализуемого функционального интерфейса обращаться внутри лямбда-выражения запрещено.

# Задачи с лямбда выражениями

Отсортировать список строк

|  |  |
| --- | --- |
| 1. | Collections.sort(list, (a, b) -> a.compareTo(b)); |

# Ссылка на метод

Применяется, если существует метод, который может решить нашу задачу.

Ссылка передается в виде:  
∙ имя\_класса::имя\_статического\_метода (для статического метода);  
∙ объект\_класса::имя\_метода (для метода экземпляра);  
∙ название\_класса::new (для конструктора).

|  |  |
| --- | --- |
| 1.  2.  3.  4.  5.  6.  7.  8. | public interface Measurable {  public int length(String string);  }  public static void main(String[] args) {  Measurable a = String::length;  System.out.println(a.length("abc"));  } |

Этот способ предпочтительнее использования лямбда-выражения (потенциально более эффективны и приставляют компилятору более качественную информацию о типе).