**Лямбда-выражение** или просто лямбда в Java — упрощённая запись анонимного класса, реализующего функциональный интерфейс.

Лямбда-выражение не выполняется само по себе, а образует реализацию метода, определенного в функциональном интерфейсе.

Лямбда-выражение состоит из двух частей: списка аргументов и тела, разделенных при помощи ->.

(a, b) -> a + b;

Типы лямбда вырожений:

* Однострочные (Одиночные)
* Блочные

Обрамляется в {}. Внутри можно использовать внутренние вложенные блоки, циклы, конструкции if, switch, создавать переменные и т. д. Если лямбда-выражение должно возвращать значение, то явно прописывается return.

Внутри лямбда-вырожений можно ссылаться на

* неизменяемые (*effectively final* - не обязательно помеченные как final) локальные переменные;
* поля класса;
* статические переменные.

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

В качестве реализации функционального интерфейса можно передовать ссылку на уже существующий метод (механизм *instance method reference*). Такая ссылка передается в виде:

* имя\_класса::имя\_статического\_метода **для статического метода**;
* объект\_класса::имя\_метода **для метода экземпляра**;
* название\_класса::new **для конструктора**.

**ФУНКЦИОНАЛЬНЫЙ ИНТЕРФЕЙС**

Если интерфейс в Java содержит один и только один абстрактный метод, то он называется **функциональным**. Этот единственный метод определяет назначение интерфейса.

Например, интерфейс Runnable из пакета java.lang.

Аннотация *FunctionalInterface*, указывает компилятору, что интерфейс функциональный (не позволяет иметь более одного абстрактного метода). Но это не обязательное условие, так как JVM считает функциональным любой интерфейс с одним абстрактным методом.

В Java 7, функциональные интерфейсы рассматривались как Single Abstract Methods (SAM). SAM обычно реализовывались с помощью анонимных классов.