Курс ЯЗЫК JAVA Курс **WEB РАЗРАБОТКА** Успеваемость Процесс обучения Курс БАЗЫ ДАННЫХ Тестирование

Контакты

Лекция 8.2: Новое в Java 8 / Лямбды (Project Lambda).

```
1. Синтаксис лямбд
                                                                                                         Приветствуем, pafnat
....1.1 Базовый синтаксис
                                                                                                           Настройки профиля
....1.2 Продвинутые возможности синтаксиса
....1.3 Вывод типов для лямбд
                                                                                                           Выйти
....1.4 Когда вывод типов для лямбд НЕ работает
2. Замыкание, захват окружения
....2.1 Effectively final
                                                                                                         Ваша успеваемость
....2.2 this/super in lambda
....2.3 Не создают всоре
                                                                                                           Java Core
                                                                                                                                    14.34%
3. Как реализованы лямбды на уровне байткода/JVM
                                                                                                            1. Основы Java
                                                                                                                                   88.07%
....3.1 Инструкция invokedynamic
                                                                                                                                   24.13%
                                                                                                            2. Базовые алгоритмы
....3.2 Поддержка языков с динамической типизацией со стороны JVM
                                                                                                            3. Исключения
                                                                                                                                    31.21%
Тесты по всей лекции
                                                                                                            4. Ввод/вывод
                                                                                                                                     0%
                                                                                                                                     0%
                                                                                                            5. Многопоточность
<u>Видео</u>
                                                                                                                                     0%
                                                                                                            6. Коллекции
<u>Лабораторные</u>
                                                                                                            7. ООП: Синтаксис
                                                                                                                                     0%
....java8.lambda.arg_shift
                                                                                                            8. ООП: Шаблоны
                                                                                                                                     0%
....java8.lambda.curring
                                                                                                            9. Продвинутые
                                                                                                                                     0%
....java8.lambda.uncurring
                                                                                                          возможности
Литература
                                                                                                            10. Java 8
                                                                                                                                     0%
```

```
interface IntegerMath {
    int operation(int a, int b);
```

1. Синтаксис лямбд

РАЗДЕЛ НАХОДИТСЯ В РАЗРАБОТКЕ

```
public class App {
5
         public static void main(String[] args) {
6
              IntegerMath add = (a,b) \rightarrow a + b;
              IntegerMath sub = (a,b) \rightarrow a - b;
```

```
8
               IntegerMath mul = (a,b) \rightarrow a * b;
  9
               IntegerMath div = (a,b) \rightarrow a / b;
 10
 11
Ну и мы и раньше это могли, на самом деле (с анонимными классами)
     public class App {
 2
          public static void main(String[] args) {
 3
              IntegerMath add = new IntegerMath() {public int operation(int a, int b) {ret
 4
              IntegerMath sub = new IntegerMath() {public int operation(int a, int b) {ret
 5
              IntegerMath mul = new IntegerMath() {public int operation(int a, int b) {ret
              IntegerMath div = new IntegerMath() {public int operation(int a, int b) {ret
```

6

```
Или даже с именованными классами
      public class App {
  2
          public static class Add implements IntegerMath {public int operation(int a, int
  3
          public static class Sub implements IntegerMath {public int operation(int a, int
  4
          public static class Mul implements IntegerMath {public int operation(int a, int
  5
          public static class Div implements IntegerMath {public int operation(int a, int
  6
  7
          public static void main(String[] args) {
  8
               IntegerMath add = new Add();
  9
               IntegerMath sub = new Sub();
 10
               IntegerMath mul = new Mul();
 11
               IntegerMath div = new Div();
 12
 13
1.1 Базовый синтаксис
```

Quiz"

Тест

1.2 Продвинутые возможности синтаксиса

РАЗДЕЛ НАХОДИТСЯ В РАЗРАБОТКЕ

РАЗДЕЛ НАХОДИТСЯ В РАЗРАБОТКЕ

Тест

8

// (Function & Serializable) // () -> () -> () -> null; // curring, uncurring

Для прохождения теста по теме TODO_java8.lambda.value-SCP (уровень сложности теста: basic) нажмите "Start

Для прохождения теста по теме TODO java8.lambda.value-F (уровень сложности теста: basic) нажмите "Start Quiz"

Тест Для прохождения теста по теме TODO_java8.labmda.curring (уровень сложности теста: mid) нажмите "Start Quiz"

Тест

Для прохождения теста по теме TODO java8.lambda.curring-2and1 (уровень сложности теста: mid) нажмите "Start Quiz"

1.4 Когда вывод типов для лямбд НЕ работает

1.3 Вывод типов для лямбд

РАЗДЕЛ НАХОДИТСЯ В РАЗРАБОТКЕ

РАЗДЕЛ НАХОДИТСЯ В РАЗРАБОТКЕ

2. Замыкание, захват окружения

РАЗДЕЛ НАХОДИТСЯ В РАЗРАБОТКЕ 2.1 Effectively final

РАЗДЕЛ НАХОДИТСЯ В РАЗРАБОТКЕ

2.2 this/super in lambda

РАЗДЕЛ НАХОДИТСЯ В РАЗРАБОТКЕ

2.3 Не создают всоре

РАЗДЕЛ НАХОДИТСЯ В РАЗРАБОТКЕ

3. Как реализованы лямбды на уровне байткода/JVM РАЗДЕЛ НАХОДИТСЯ В РАЗРАБОТКЕ

3.1 Инструкция invokedynamic

РАЗДЕЛ НАХОДИТСЯ В РАЗРАБОТКЕ

РАЗДЕЛ НАХОДИТСЯ В РАЗРАБОТКЕ

3.2 Поддержка языков с динамической типизацией со стороны JVM

Тесты по всей лекции

Тест, состоящий из случайных вопросов тестов этой лекции

Lambda, часть изменениям Stream API)

[INFO] Scanning for projects...

DBB1 Guilding Lambda-text 1.0-MapShOT

Сергей Куксенко и Алексей Шипилёв, "Через тернии к лямбдам, часть 1"

Execut) To not the full stack trace of the errors, re-ron Mayon with the -e smitch.

DROOM) Re-run Maxen using the -X match to enable full debug logging.

Сложное НЕ мое видео

ΤΕCΤΑ ΠΟΚΑ ΗΕΤ

assort Failed to execute goal org spacks mayor plugins mayor compiler -plugin: 2. 8.2/compile (default-compile) on project 1 MODEL-TEST: Compiliation failure (MUDE) Fasture executing javac, but could not parse the error: EMMON! parac: invalid target release: 1,8 EMBOR: Usage: pavas exptions: recurse files: EMBOR: use -help for a list of possible options ERRORS -> (MAJE 1)

}

>> was: 123 >> now: 312

ger>>>>

ger>>>>

import java.util.function.*;

public class FunctionUtils {

BiFunction<T, T, BiFunction<T, T, T>> f) {

import java.util.function.Function;

public static void main(String[] args) {

Видео

(ARCH] For more information about the errors and possible solutions, please read the following articles: teach; (Help 1) http://cmaki.apache.org/confluence/display/MAVEN/MojsFailureEsception ed Oct 17, 18:47:23 shadeeps continue -/mickspace-Lad ed Oct 17, 18:47:27 Hundriger regimes -/Harbysco-Landda/Landda-Yout & gaport JAVA Mike-Jtrumks/Landda/Butlifysku-Landda-GRZ1017-Wefaultr wed Sct 17, 19:47:22 shadospersophore - Corkspace Landstaff, adult text 5 export PATH-5:AVA HOME/Exm:5PATH
wed Sct 17, 19:47:23 shadospersophore - Corkspace Landstaff, adult text 8 mon clean install - DakinTexts-true

AMBING) Some problems were encountered while building the offective model for comparationlesses and included and a second sections and the comparation of the compara assessed "build plugins plugin version" for erg spacks maven plugins maven computer plugin is missing @ line #8, Interes

saming) it is highly recommended to fix these problems because they threaten the stability of your build.

HARMING) For this reason, future Maren versions might no longer support building such malformed projects.

Набор август 2013, вторая дополнительная лекция (часть лекции посвящена изменению методов, часть – Project

```
0:00 / 1:27:19
Лабораторные
java8.lambda.arg_shift
java8.lambda.arg_shift
Реализуйте функцию высшего порядка, которая делает "циклический shift вправо аргументов" для каррированной
функции трех аргументов Function<T, Function<T, Function<T, T>>>.
Пример
   import java.util.function.Function;
   public class App {
       public static void main(String[] args) {
           Function<Integer, Function<Integer, Function<Integer, Integer>>>
```

was = $x \rightarrow y \rightarrow z \rightarrow 100 * x + 10 * y + z;$

now = FunctionUtils.shift(was);

Function<Integer, Function<Integer, Function<Integer, Integer>>>

System.out.println("was: " + was.apply(1).apply(2).apply(3)); System.out.println("now: " + now.apply(1).apply(2).apply(3));

```
public class FunctionUtils {
       public static <T> Function<T, Function<T, Function<T, T>>> shift(Function<T, Fun</pre>
   ction<T, Function<T, T>>> f) {
java8.lambda.curring
java8.lambda.curring
Реализовать функцию высшего порядка, которая осуществляет преобразование между BiFunction<T, T,
BiFunction<T, T, T>> и Function<T, Function<T, Function<T, Function<T, T>>>> полностью сохраняя функционал.
То есть
   import java.util.function.*;
   public class App {
```

```
now = FunctionUtils.curring(was);
        System.out.println("was: " + was.apply(1, 2).apply(3, 4));
        System.out.println("now: " + now.apply(1).apply(2).apply(3).apply(4));
   }
}
>> was: 1234
>> now: 1234
```

public static <T> Function<T, Function<T, Function<T, Function<T, T>>>> curring(

BiFunction<Integer, Integer, BiFunction<Integer, Integer>>

was = $(x, y) \rightarrow (z, t) \rightarrow 1000 * x + 100 * y + 10 * z + t;$

Function<Integer, Function<Integer, Function<Integer, Inte

```
java8.lambda.uncurring
java8.lambda.uncurring
Реализовать функцию высшего порядка, которая осуществляет преобразование между Function<T, Function<T,
Function<T, Function<T, T>>>> и BiFunction<T, T, BiFunction<T, T, T>> полностью сохраняя функционал.
   import java.util.function.*;
   public class App {
       public static void main(String[] args) {
```

Function<Integer, Function<Integer, Function<Integer, Inte

was = $x \rightarrow y \rightarrow z \rightarrow t \rightarrow 1000 * x + 100 * y + 10 * z + t;$

BiFunction<Integer, Integer, BiFunction<Integer, Integer>>

System.out.println("was: " + was.apply(1).apply(2).apply(3).apply(4));

now = FunctionUtils.unCurring(was);

System.out.println("now: " + now.apply(1, 2).apply(3, 4));

```
}
>> was: 1234
>> now: 1234
import java.util.function.*;
public class FunctionUtils {
    public static <T> BiFunction<T, T, BiFunction<T, T, T>> unCurring(Function<T, Fu</pre>
nction<T, Function<T, Function<T, T>>>> f) {
```

```
Кратко от Oracle
oracle.com/JavaTutorial: "Lambda Expressions"
```

© 2014 Golovach Courses |

Литература

```
Angelika Langer: "Lambda Expressions in Java. Reference"
Brian Goetz объясняет во что компилируются лямбда-выражения. Сложная статья, желательно понимание JSR
292 (invokedynamic байткод)
```

Angelika Langer: "Lambda Expressions in Java. Tutorial"

```
"Translation of Lambda Expressions"
Maurice Naftalin's "Lambda FAQ"
```

Tutorial (попроще) и Reference (посложнее) от Angelika Langer

Ted Neward, "Java 8: Lambdas, Part 1" Ted Neward, "Java 8: Lambdas, Part 2"