1. Try-with-resources 2. Multi-catch 3. More presice rethrow Тесты по всей лекции Видео Лабораторные ex.java7.try-with-resources-2	Настройки профиля Выйти Ваша успеваемост	
<u>ex.java7.try-with-resources-3</u> Литература . Try-with-resources	Java Core 1. Основы Java 2. Базовые алгор 3. Исключения 4. Ввод/вывод	31.21% 0%
GECT ля прохождения теста по теме Ex.TWR.One.Strange (уровень сложности теста: Basic) нажмите "Start Quiz" Да тестах используется класс Ok – это AutoCloseable, который не выбрасывает никаких исключений.	5. Многопоточнос 6. Коллекции 7. ООП: Синтакси 8. ООП: Шаблоны алее 9. Продвинутые возможности 10. Java 8	0% c 0%
<pre>public class Ok implements AutoCloseable { private String msg; public Ok(String msg) { this.msg = msg; System.err.println("new:" + msg); }</pre>	To. Sava o	070
<pre>public void close() { System.err.println("close:" + msg); } </pre>		
ест ля прохождения теста по теме Ex.Java8.TWR.Two-Independent (уровень сложности теста: Mid) нажмите "Star uiz" Далее в тестах используется класс FailClose	rt	
<pre>public class FailClose implements AutoCloseable { private String msg; public FailClose(String msg) { this.msg = msg; System.err.println("new:" + msg); } public void close() {</pre>		
<pre>System.err.println("close:" + msg); throw new Error(msg); } </pre>		
ro AutoCloseable, который выбрасывает Error при вызове close(). CCT ля прохождения теста по теме Ex.Java8.TWR.Two-Dependent (уровень сложности теста: Mid) нажмите "Start uiz" Далее в тестах используется классы Ok, FailCreate, FailClose Ok – это AutoCloseable, который не ыбрасывает никаких исключений. FailCreate – это AutoCloseable, который выбрасывает Error при вызове		
риblic class Ok implements AutoCloseable { private String msg; public Ok(String msg) { this.msg = msg;		
<pre>System.err.println("new:" + msg); } public void close() { System.err.println("close:" + msg); }</pre>		
<pre>public class FailCreate implements AutoCloseable { private String msg; public FailCreate(String msg) { this.msg = msg; System.err.println("new:" + msg); throw new Error(msg);</pre>		
<pre>public void close() { System.err.println("close:" + msg); } public class FailClose implements AutoCloseable {</pre>		
<pre>private String msg; public FailClose(String msg) { this.msg = msg; System.err.println("new:" + msg); } public void close() { System.err.println("close:" + msg);</pre>		
throw new Error(msg); } Java 7 в пакете java.lang появился следующий интерфейс		
<pre>package java.lang; /** A resource that must be closed when it is no longer needed. * A author Josh Bloch</pre>	?	
<pre>* @since 1.7 * public interface AutoCloseable { void close() throws Exception; } ля вспомогательных целей далее будут использоваться 3 реализации этого интерфейса: Ok, FailClose,</pre>		
ailCreate. ok — pecypc (AutoCloseable), который создается и закрывается без возникновения исключительных ситуаций. public class Ok implements AutoCloseable { private String msg; public Ok(String msg) { this.msg = msg;	?	
<pre>System.err.println("new:" + msg); public void close() { System.err.println("close:" + msg); } } </pre>		
ailClose — ресурс (AutoCloseable), который без возникновения исключительных ситуация создается, но при акрытии выбрасывает исключительную ситуацию. public class FailClose implements AutoCloseable { private String msg; public FailClose(String msg) { this.msg = msg;	?	
<pre>this.msg = msg; System.err.println("new:" + msg); public void close() { System.err.println("close:" + msg); throw new Error(msg); }</pre>		
3 3 3 3 3 3 3 3 3 3	?	
<pre>private String msg; public FailCreate(String msg) { this.msg = msg; System.err.println("new:" + msg); throw new Error(msg); } public void close() { System.err.println("close:" + msg);</pre>		
9 System.err.println("close:" + msg); 10 } 11 } k: создается, заходим в тело, закрывается. 1 public class App { 2 public static void main(String[] args) {	?	
<pre>try (Ok x = new Ok("x")) {</pre>		
<pre>9</pre>	?	
<pre>public static void main(String[] args) { try (FailClose x = new FailClose("x")) { System.err.println("body"); } }</pre>		
<pre>9</pre>		
<pre>public class App { public static void main(String[] args) { try (FailCreate x = new FailCreate("x")) { System.err.println("body"); } }</pre>	?	
<pre>6 } 7 } 8 9 >> new:x 10 >> Exception in thread "main" java.lang.Error: x</pre> Tor: по поведению, try-c-одним-ресурсом эквивалентен следующему коду		
<pre>public class App { public static void main(String[] args) { AutoCloseable x = new try { System.err.println("body"); } finally {</pre>	?	
7		
аботает исключительно для реализаций интерфейса AutoCloseable public class App { public static void main(String[] args) { try (Integer k = new Integer(0)) { System.err.println("body"); }	?	
6 } 7 } 8 9 >> не компилируется еременные инициализированные в круглых скобках "эффективно финальные" (не допускают повторного рисвоения)		
<pre>public class App { public static void main(String[] args) { try (Ok x = new Ok("x")) { x = null; } }</pre>	?	
<pre>6 } 7 } 8 9 >> не компилируется бласть видимости переменных — тело try 1 public class App {</pre>	2	
<pre>public static void main(String[] args) { try (Ok x = new Ok("x")) {</pre>		
<pre>8 } 9 10 >> не компилируется 1 public class App { 2 public static void main(String[] args) { 3 try (Ok x = new Ok("x")) { 4 //</pre>	?	
<pre>5</pre>		
<pre>public class App { public static void main(String[] args) { try (Ok x = new Ok("x")) { // } finally { AutoCloseable ac = x; } }</pre>	?	
7		
ривіс class App { public static void main(String[] args) { Ok x = null; try (x = new Ok("x")) { System.err.println("body");	?	
System.eff.pffmtfm(body), AutoCloseable ac = x;		
аботает всего пара "странных" конструкций. Вызвать close() самому (компилятор не защищает), в таком случае будет ДВА закрытия вашего ресурса рublic class App { public static void main(String[] args) { try (Ok x = new Ok("x")) {	?	
<pre>4</pre>		
>> body >> close:x >> close:x ли вот такой трюк (компилятор требует, что бы ресурс был объявлен и инициализирован в "круглых скобках", е требует, что бы он был СКОНСТРУИРОВАН в них)	, НО	
<pre>public class App { public static void main(String[] args) { Ok tmp = new Ok("x"); try (Ok x = tmp) { System.err.println("body"); } AutoCloseable ac = tmp; }</pre>	?	
<pre>7 AutoCloseable ac = tmp; 8 } 9 } 10 11</pre>		
дин ресурс + finally + catch y-with-resources 2. Multi-catch		
1 × 8. More presice rethrow	?	
ест ля прохождения теста по теме Ex.MorePreciseRethow (уровень сложности теста: Mid) нажмите "Start Quiz" ест ля прохождения теста по теме Ex.Java8.TWR.ByHands нажмите "Start Quiz"		
ест, состоящий из случайных вопросов тестов этой лекции ome issues occured with quiz randomizer :(
Видео ?? Набор декабрь 2013		
?? Набор октябрь 2013 ?? Набор июль 2013 ?? Набор апрель 2013 ?? Набор февраль 2013 ?? Набор январь 2013 ?? Набор октябрь 2012 абор июль 2013: лекция #4 (с части 4 до конца)		
абор апрель 2013: лекция #5 (части 4 до конца) абор апрель 2013: лекция #5 (части 1+2+3, потом уже Java I/O) абор февраль 2013: лекция #4 (РЕКОМЕНДУЕМОЕ) абор январь 2013: лекция #4 – https://www.youtube.com/watch? =6T67RHW4O1c&index=3&list=PLoij6udfBncgrrMY5sKe-V2PxXFk5qfkz – ЧТО ЭТО???		
laбораторные utoCloseableFactory – это "заменитель" конструктора некоторого AutoCloseable		
public interface AutoCloseableFactory { public AutoCloseable create() throws Throwable; } ryBody — это "заменитель" "тела try"	?	
public interface TryBody { public void runBody() throws Throwable; } ример: Один ресурс (с решением) ешение на Java 7 С применением try-with-resources		
public class AnswerOneResourceJava7 { public static void call(AutoCloseableFactory factory, TryBody body) throws The try (AutoCloseable resource = factory.create()) { body.runBody(); }	hrow	
<pre>public class AnswerOneResourceJava6 { public static void call(AutoCloseableFactory factory, TryBody body) throws</pre>	? Thro	
<pre>public static void call(AutoCloseableFactory factory, TryBody body) throws // create A AutoCloseable resource = factory.create(); // run body try { body.runBody(); } catch (Throwable bodyEx) { try {</pre>		
<pre>10</pre>		
resource.close(); 18 } 19 } Вабораторная: ex.java7.try-with-resources-2		
ешение на Java 7 С применением try-with-resources 1	?	
TryBody body) throws Throwable { try (AutoCloseable a = factoryA.create(); AutoCloseable b = factoryB.create()) { body.runBody(); }		
12 } 13 } адание: реализовать эту логику БЕЗ использования try-with-resources. абораторная: ex.java7.try-with-resources-3		
ешение на Java 7 С применением try-with-resources 1	?	
AutoCloseableFactory factoryC, TryBody body) throws Throwable { try (AutoCloseable a = factoryA.create(); AutoCloseable b = factoryB.create(); AutoCloseable c = factoryC.create() 10 } {		
11		

телефон: +(38) 063-048-7663 | Email: GolovachCourses@gmail.com | Skype: GolovachCourses