|  |  |
| --- | --- |
|  | Министерство образования и науки Российской Федерации |
| Федеральное государственное автономное образовательное учреждение высшего образования |
| «Уральский федеральный университет имени первого Президента России Б. Н. Ельцина» |
| Институт радиоэлектроники и информационных технологий – РтФ |
|  |
|  | |

**ОТЧЕТ**

О ВЫПОЛНЕНИИ

ЛАБОРАТОНОЙ РАБОТЫ № 4

Руководитель ст. пр. Н. А. Архипов

Студент гр. РИМ-130971 Е.В. Треглазов

Екатеринбург 2024

Репозиторий: https://github.com/kuk86kuk/java\_ufru\_2024.

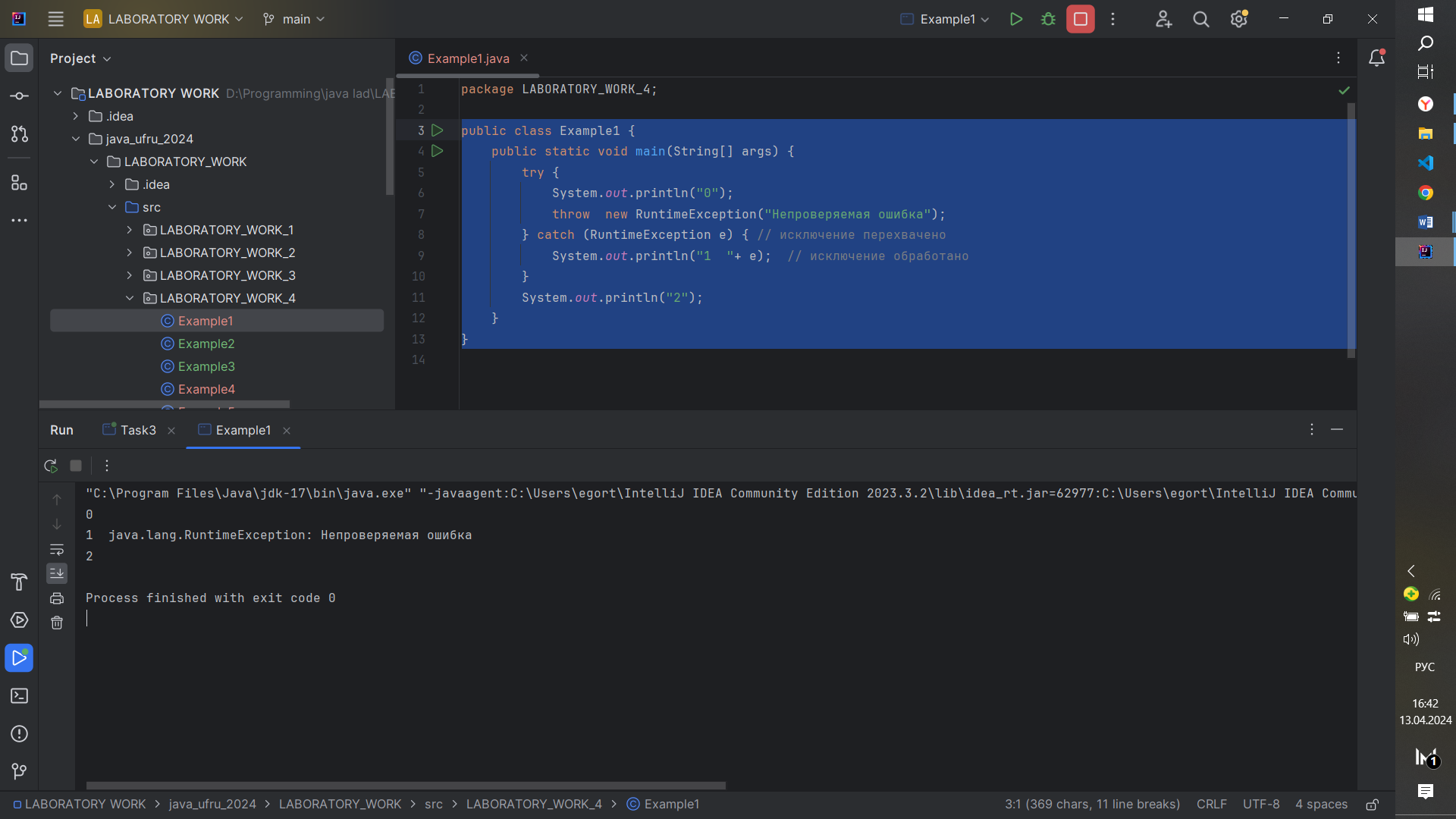
**Цель:** знакомство с иерархией классов исключений и получение навыков обработки ошибок.

Example1

public class Example1 {  
 public static void main(String[] args) {  
 try {  
 System.*out*.println("0");  
 throw new RuntimeException("Непроверяемая ошибка");  
 } catch (RuntimeException e) { // исключение перехвачено  
 System.*out*.println("1 "+ e); // исключение обработано  
 }  
 System.*out*.println("2");  
 }  
}

Код выполняет следующие действия:

* Выводит на экран строку "0".
* Вызывает оператор throw, который генерирует исключение типа RuntimeException с сообщением "Непроверяемая ошибка".
* В блоке catch исключение RuntimeException перехватывается и выводится на экран сообщение "1 Непроверяемая ошибка".
* Выводит на экран строку "2".

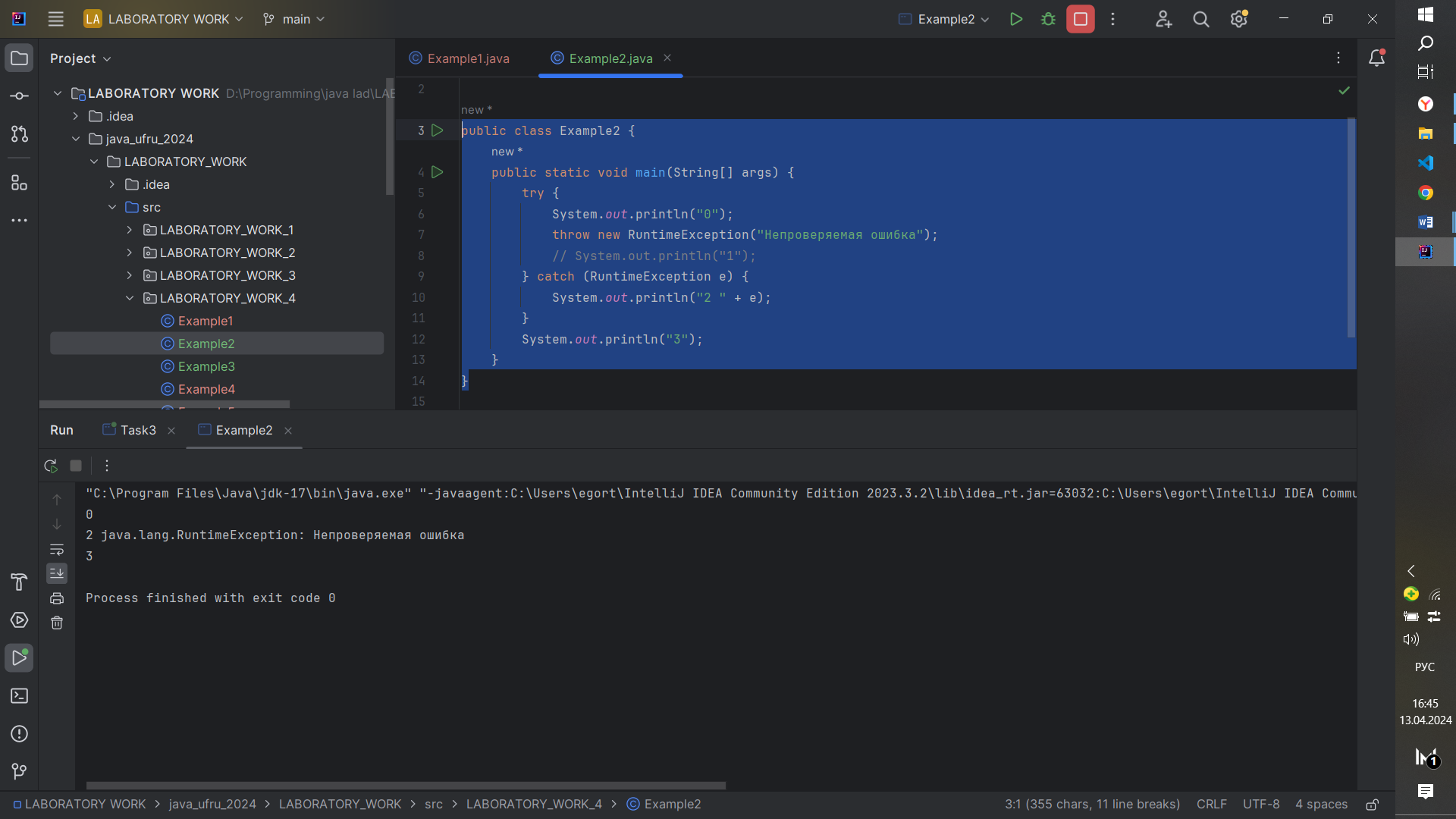


Example2

public class Example2 {  
 public static void main(String[] args) {  
 try {  
 System.*out*.println("0");  
 throw new RuntimeException("Непроверяемая ошибка");  
 // System.out.println("1");  
 } catch (RuntimeException e) {  
 System.*out*.println("2 " + e);  
 }  
 System.*out*.println("3");  
 }  
}

Код выполняет следующие действия:

* Выводит на экран строку "0".
* Вызывает оператор throw, который генерирует исключение типа RuntimeException с сообщением "Непроверяемая ошибка".
* В блоке catch исключение RuntimeException перехватывается и выводится на экран сообщение "2 Непроверяемая ошибка".
* Выводит на экран строку "3".

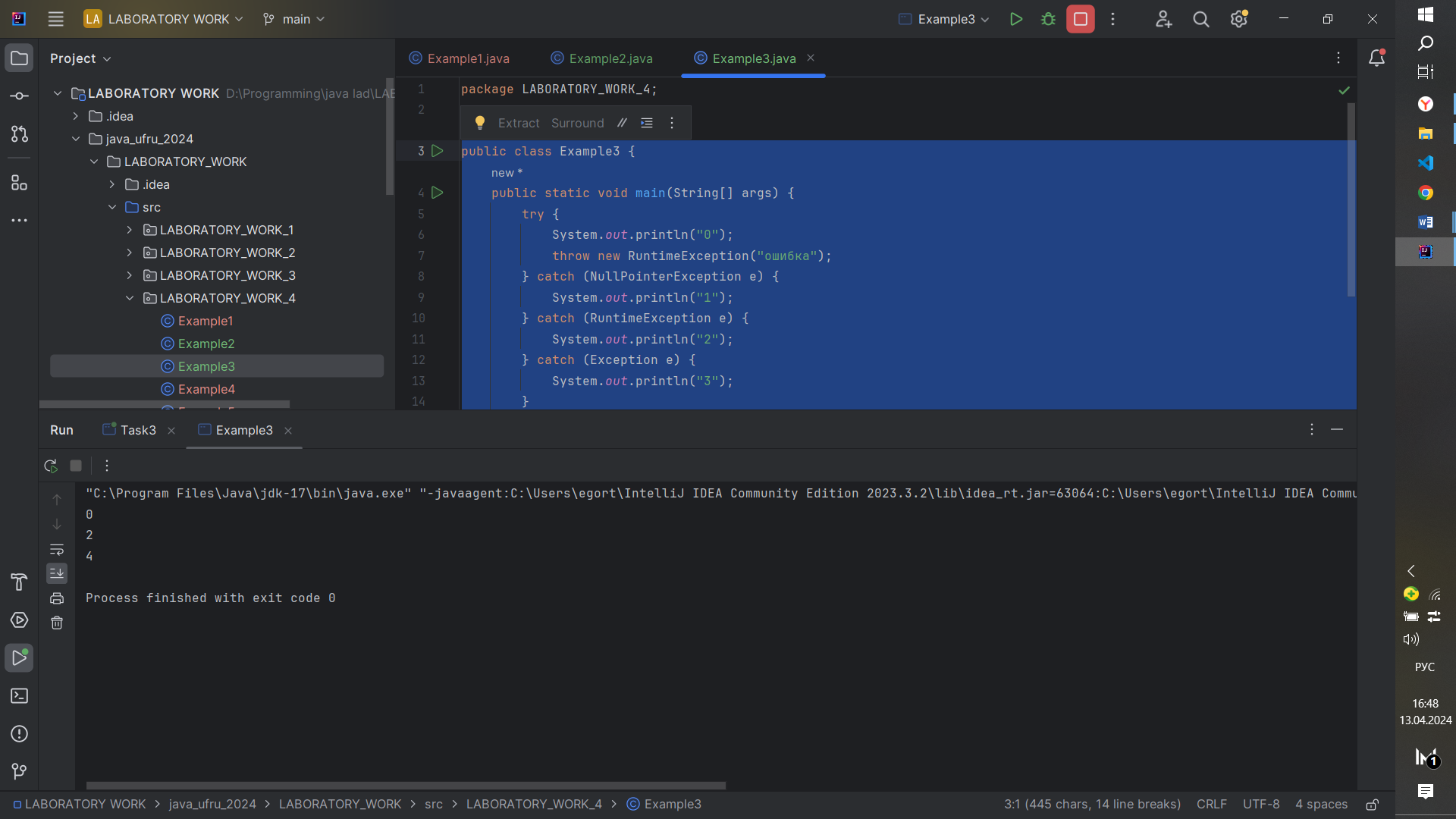


Example3

public class Example3 {  
 public static void main(String[] args) {  
 try {  
 System.*out*.println("0");  
 throw new RuntimeException("ошибка");  
 } catch (NullPointerException e) {  
 System.*out*.println("1");  
 } catch (RuntimeException e) {  
 System.*out*.println("2");  
 } catch (Exception e) {  
 System.*out*.println("3");  
 }  
 System.*out*.println("4");  
 }  
}

Код выполняет следующие действия:

* Выводит на экран строку "0".
* Вызывает оператор throw, который генерирует исключение типа RuntimeException с сообщением "ошибка".
* В блоке catch исключение RuntimeException перехватывается и выводится на экран сообщение "2".
* Выводит на экран строку "4".

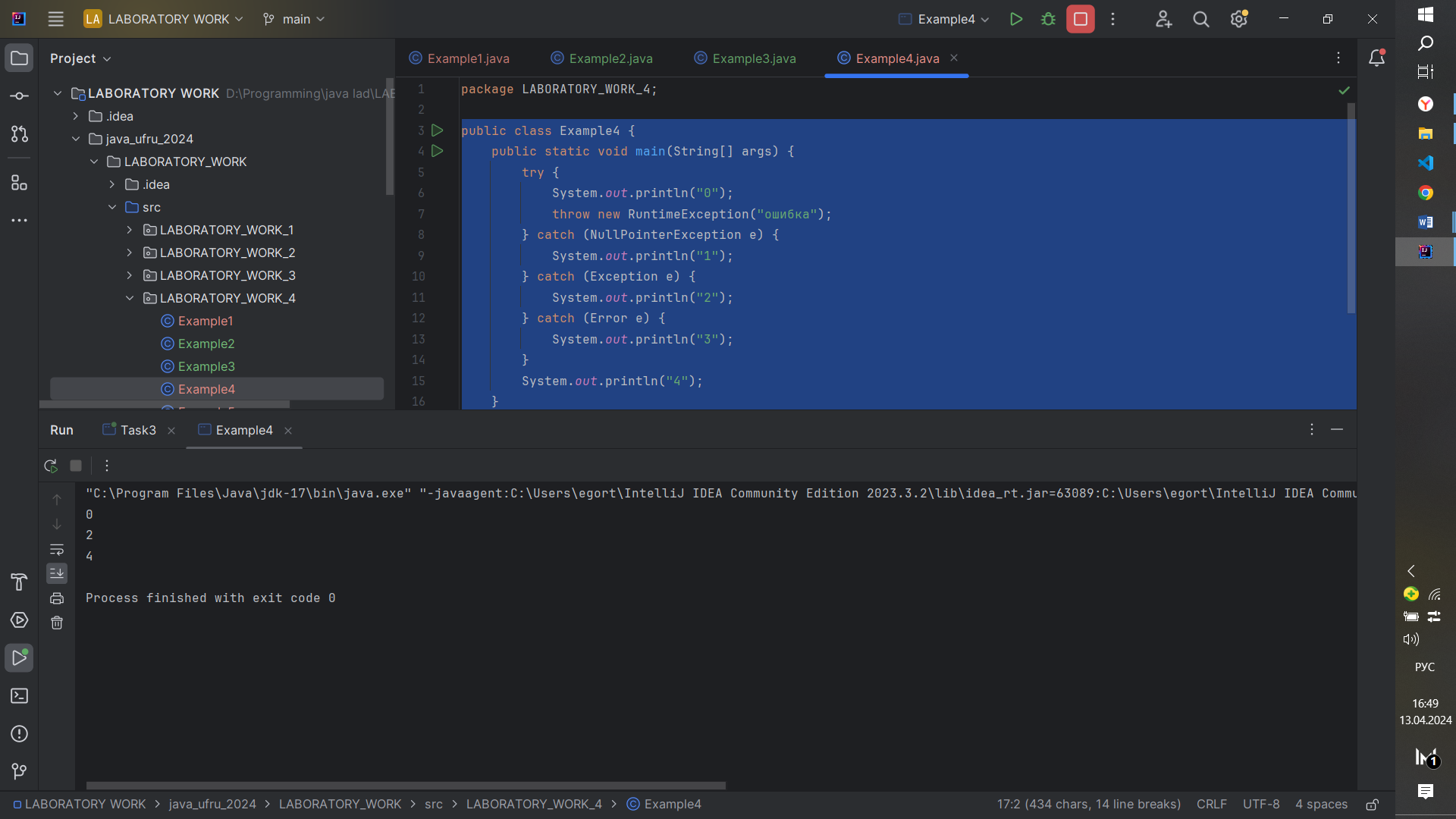


Example4

public class Example4 {  
 public static void main(String[] args) {  
 try {  
 System.*out*.println("0");  
 throw new RuntimeException("ошибка");  
 } catch (NullPointerException e) {  
 System.*out*.println("1");  
 } catch (Exception e) {  
 System.*out*.println("2");  
 } catch (Error e) {  
 System.*out*.println("3");  
 }  
 System.*out*.println("4");  
 }  
}

Код выполняет следующие действия:

* Выводит на экран строку "0".
* Вызывает оператор throw, который генерирует исключение типа RuntimeException с сообщением "ошибка".
* В блоке catch исключение RuntimeException перехватывается и выводится на экран сообщение "2".
* Выводит на экран строку "4".

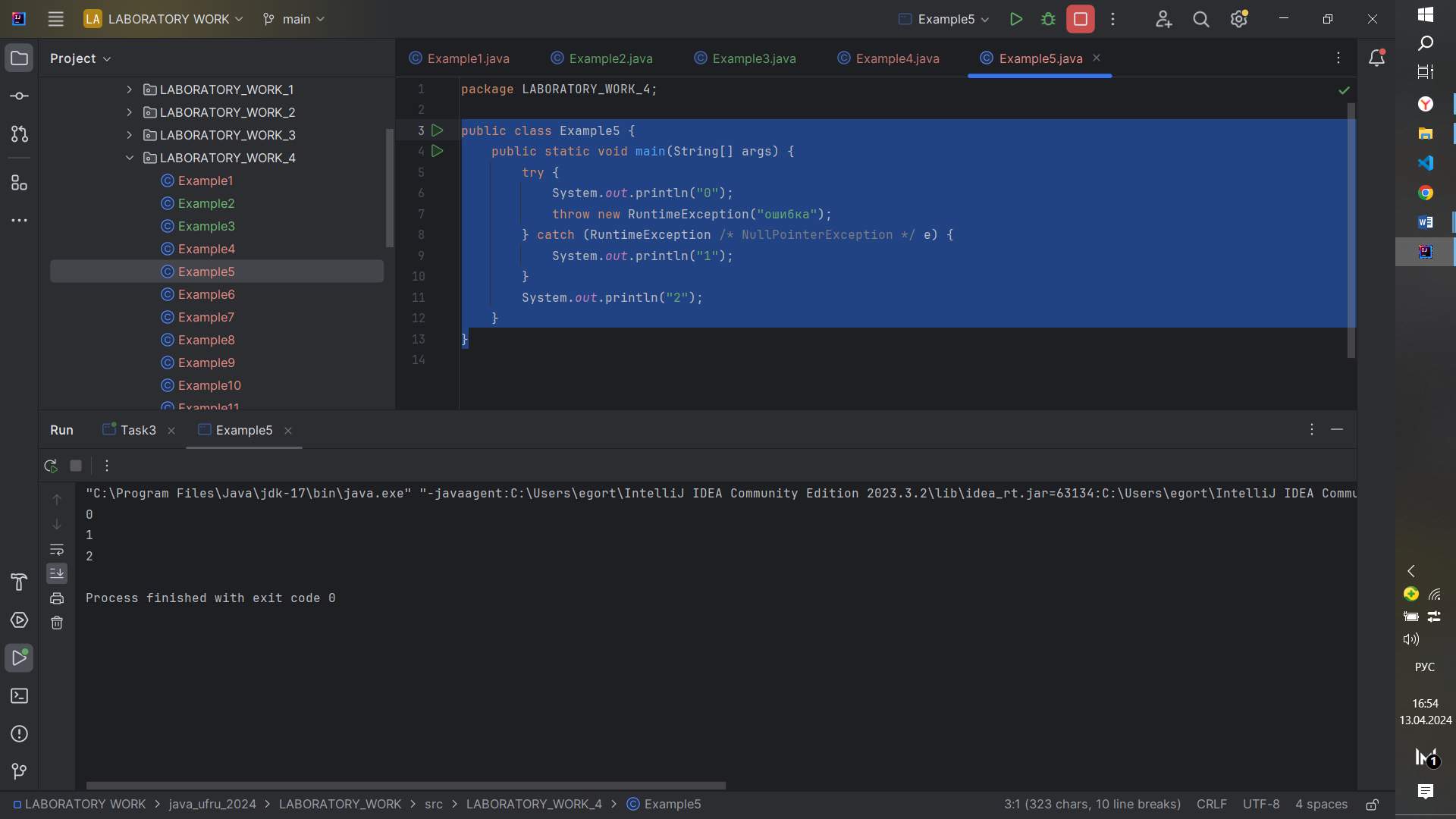


Example5

public class Example5 {  
 public static void main(String[] args) {  
 try {  
 System.*out*.println("0");  
 throw new RuntimeException("ошибка");  
 } catch (RuntimeException /\* NullPointerException \*/ e) {  
 System.*out*.println("1");  
 }  
 System.*out*.println("2");  
 }  
}

Код выполняет следующие действия:

* Выводит на экран строку "0".
* Вызывает оператор throw, который генерирует исключение типа RuntimeException с сообщением "ошибка".
* В блоке catch исключение RuntimeException перехватывается и выводится на экран сообщение "1".
* Выводит на экран строку "2".

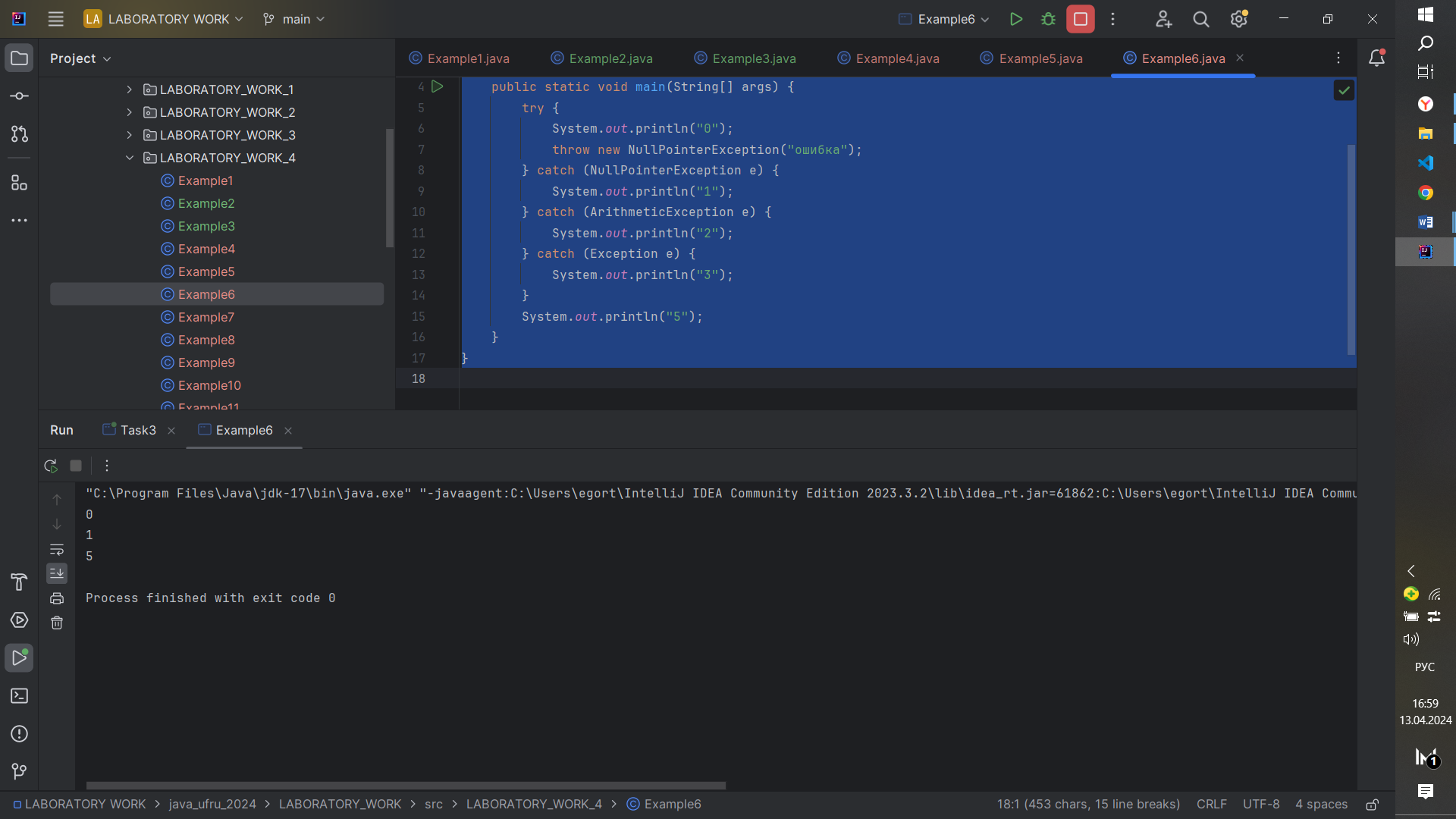


Example6

public class Example6 {  
 public static void main(String[] args) {  
 try {  
 System.*out*.println("0");  
 throw new NullPointerException("ошибка");  
 } catch (NullPointerException e) {  
 System.*out*.println("1");  
 } catch (ArithmeticException e) {  
 System.*out*.println("2");  
 } catch (Exception e) {  
 System.*out*.println("3");  
 }  
 System.*out*.println("5");  
 }  
}

Код выполняет следующие действия:

* Выводит на экран строку "0".
* Вызывает оператор throw, который генерирует исключение типа NullPointerException с сообщением "ошибка".
* В блоке catch исключение NullPointerException перехватывается и выводится на экран сообщение "1".
* Выводит на экран строку "5".

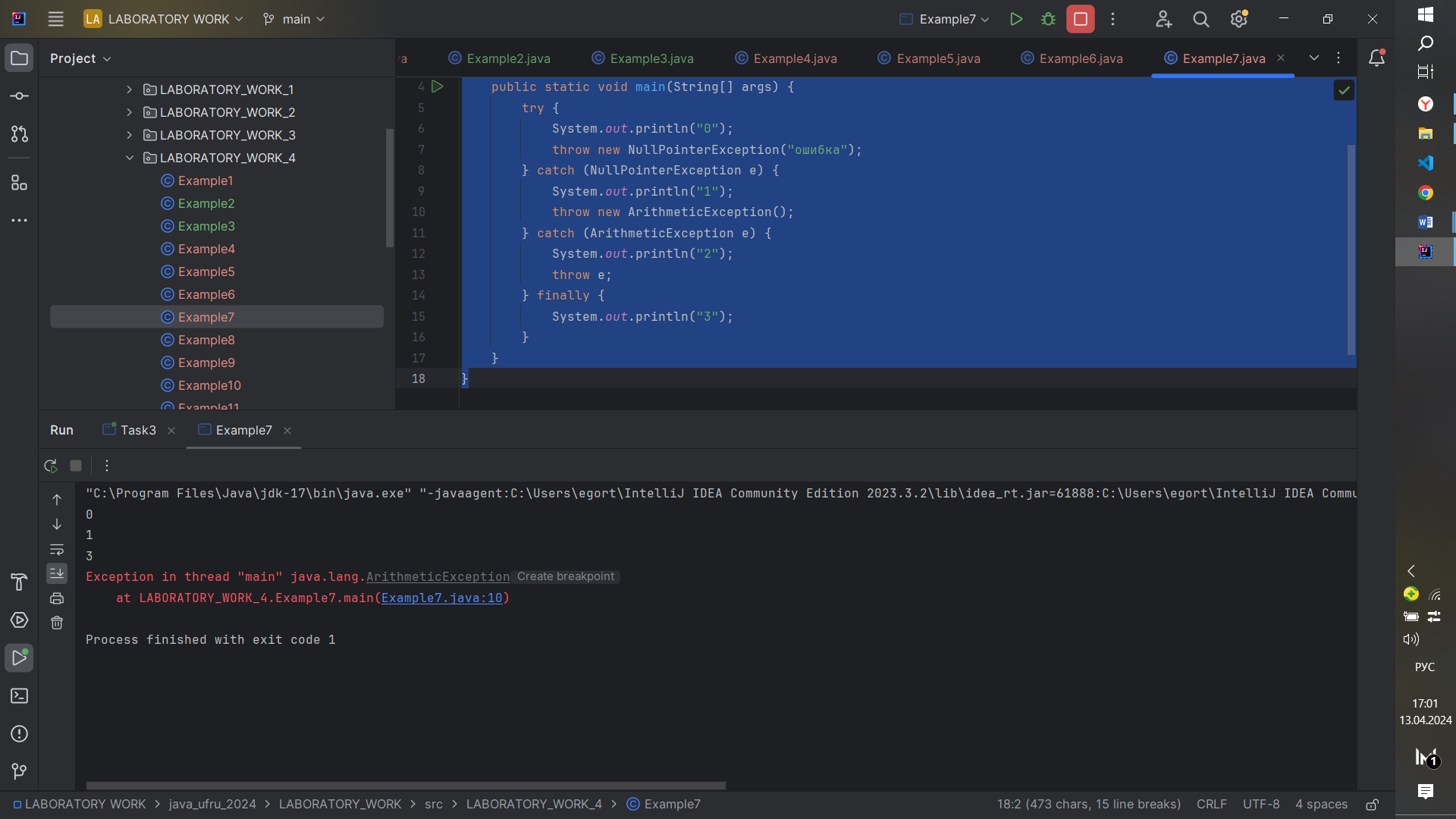


Example7

public class Example7 {  
 public static void main(String[] args) {  
 try {  
 System.*out*.println("0");  
 throw new NullPointerException("ошибка");  
 } catch (NullPointerException e) {  
 System.*out*.println("1");  
 throw new ArithmeticException();  
 } catch (ArithmeticException e) {  
 System.*out*.println("2");  
 throw e;  
 } finally {  
 System.*out*.println("3");  
 }  
 }  
}

Код выполняет следующие действия:

* Выводит на экран строку "0".
* Вызывает оператор throw, который генерирует исключение типа NullPointerException с сообщением "ошибка".
* В блоке catch исключение NullPointerException перехватывается и выводится на экран сообщение "1".
* Внутри блока catch вызывается оператор throw, который генерирует исключение типа ArithmeticException.
* В блоке catch исключение ArithmeticException перехватывается и выводится на экран сообщение "2".
* Внутри блока catch вызывается оператор throw, который перебрасывает исключение ArithmeticException.
* В блоке finally выводится на экран сообщение "3".

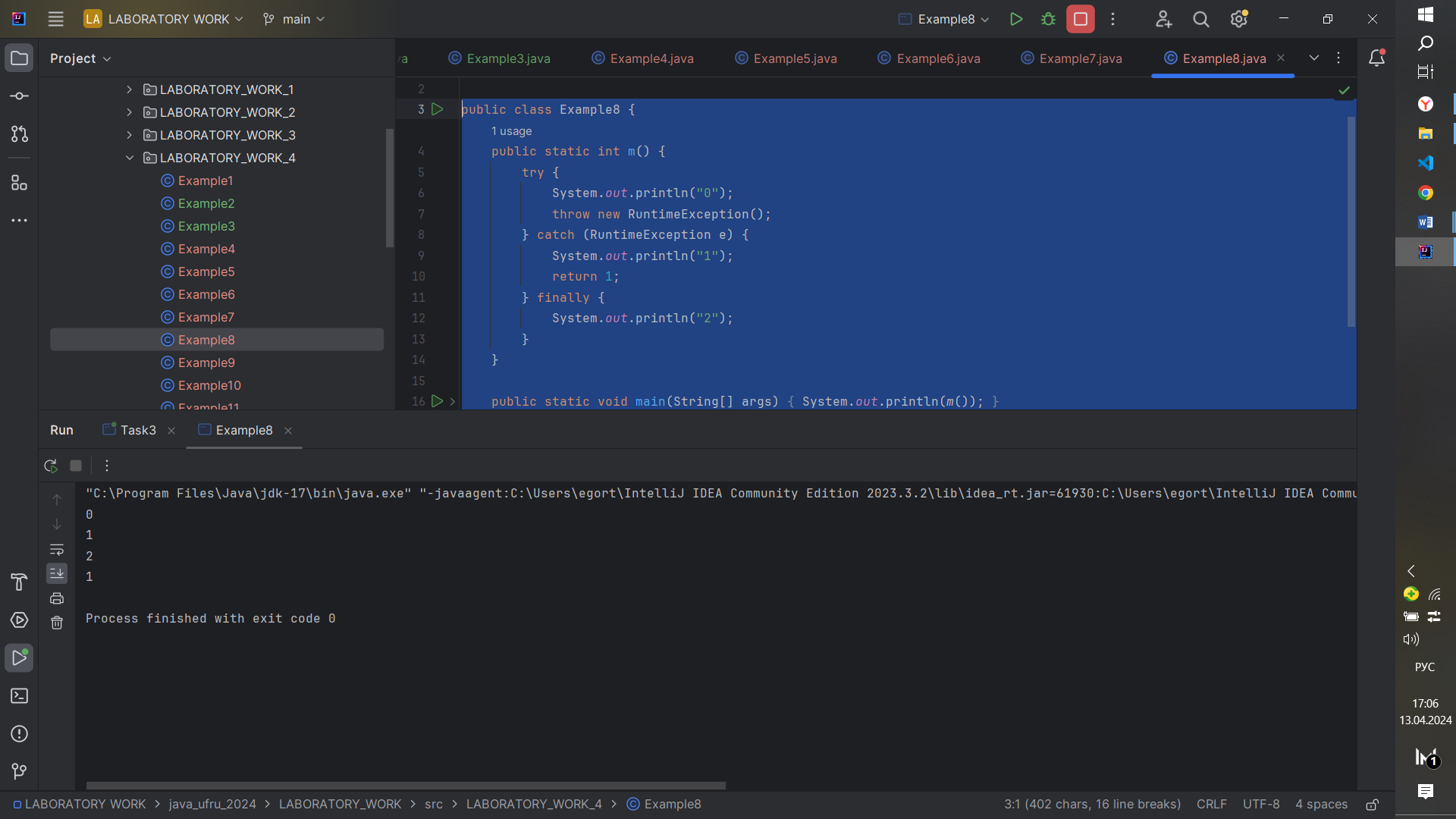


Example8

public class Example8 {  
 public static int m() {  
 try {  
 System.*out*.println("0");  
 throw new RuntimeException();  
 } catch (RuntimeException e) {  
 System.*out*.println("1");  
 return 1;  
 } finally {  
 System.*out*.println("2");  
 }  
 }  
  
 public static void main(String[] args) {  
 System.*out*.println(*m*());  
 }  
}

Код выполняет следующие действия:

* Вызывается метод m().
* Выводит на экран строку "0".
* Вызывается оператор throw, который генерирует исключение типа RuntimeException.
* В блоке catch исключение RuntimeException перехватывается и выводится на экран сообщение "1".
* В блоке catch возвращается значение 1.
* В блоке finally выводится на экран сообщение "2".
* Вызывается метод m() в методе main().
* Выводится на экран результат вызова метода m(), т.е. 1.

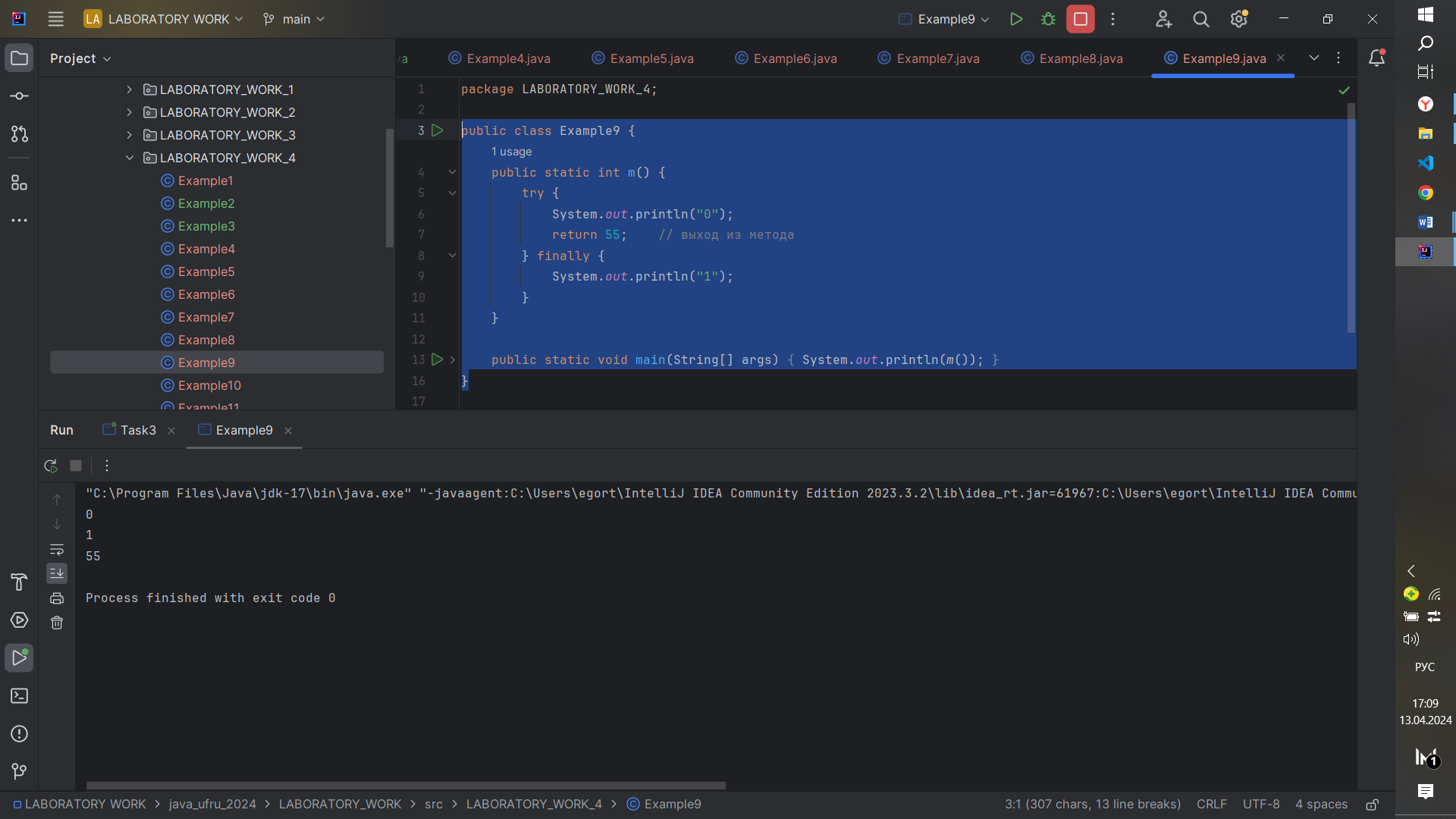


Example9

public class Example9 {  
 public static int m() {  
 try {  
 System.*out*.println("0");  
 return 55; // выход из метода  
 } finally {  
 System.*out*.println("1");  
 }  
 }  
  
 public static void main(String[] args) {  
 System.*out*.println(*m*());  
 }  
}

Код выполняет следующие действия:

* Вызывается метод m().
* Выводит на экран строку "0".
* Вызывается оператор return, который возвращает значение 55 и завершает выполнение метода m().
* В блоке finally выводится на экран сообщение "1".
* Вызывается метод m() в методе main().
* Выводится на экран результат вызова метода m(), т.е. 55.

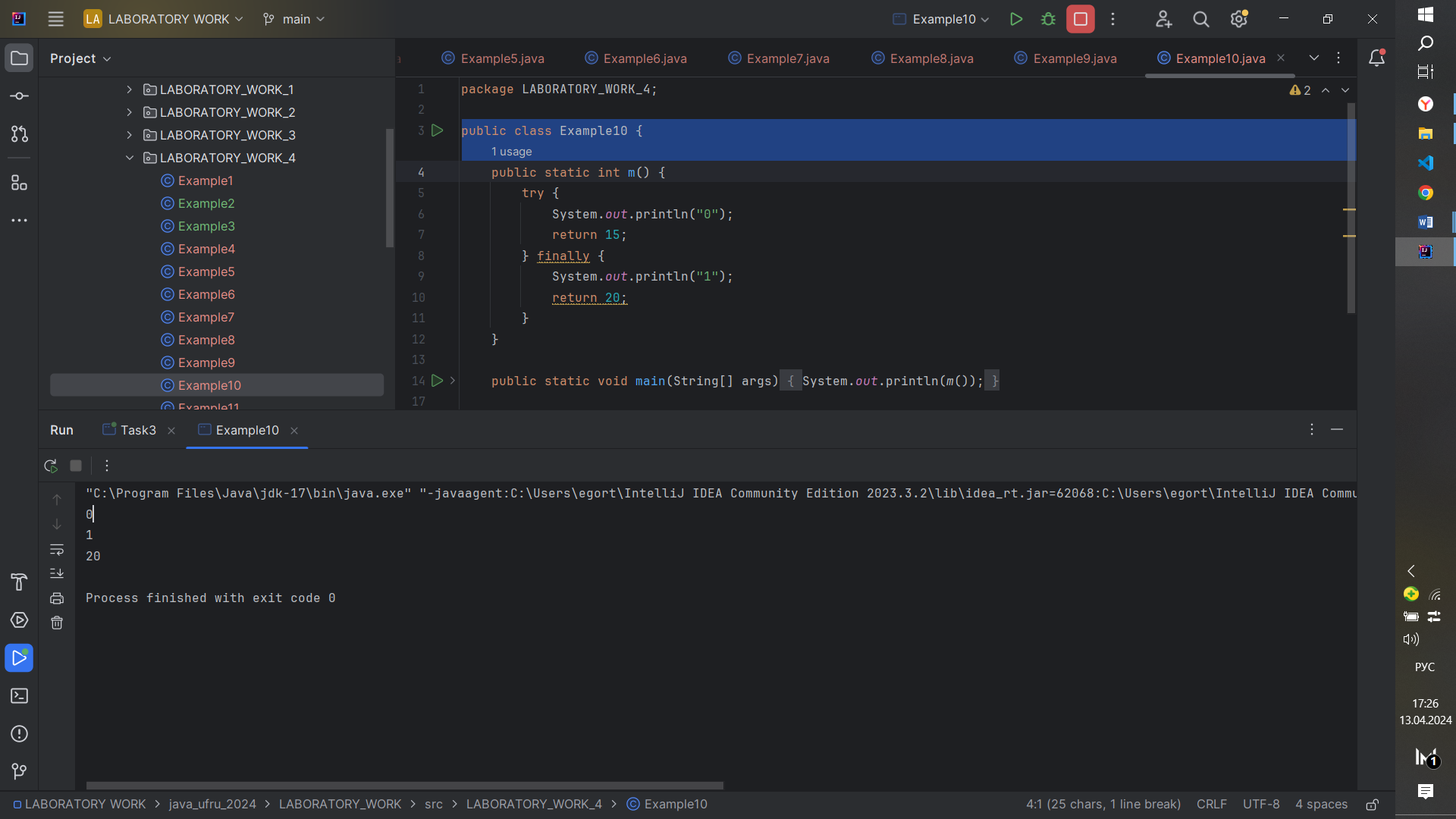


Example10

public class Example10 {  
 public static int m() {  
 try {  
 System.*out*.println("0");  
 return 15;  
 } finally {  
 System.*out*.println("1");  
 return 20;  
 }  
 }  
  
 public static void main(String[] args) {  
 System.*out*.println(*m*());  
 }  
  
}

Код выполняет следующие действия:

* Вызывается метод m().
* Выводит на экран строку "0".
* Вызывается оператор return, который возвращает значение 15 и завершает выполнение метода m().
* В блоке finally выводится на экран сообщение "1".
* В блоке finally вызывается оператор return, который возвращает значение 20.
* Вызывается метод m() в методе main().
* Выводится на экран результат вызова метода m(), т.е. 20.

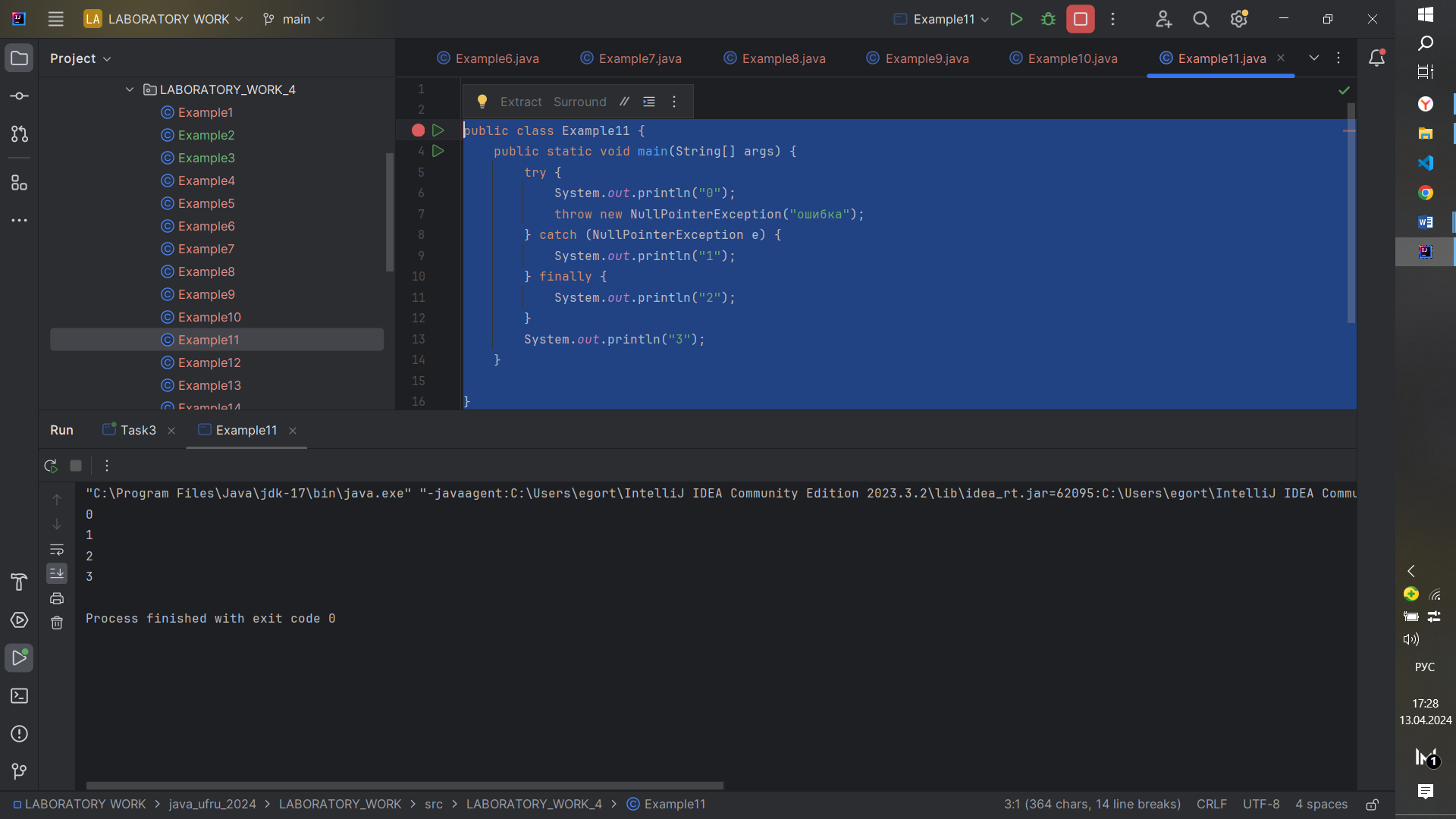


Example11

public class Example11 {  
 public static void main(String[] args) {  
 try {  
 System.*out*.println("0");  
 throw new NullPointerException("ошибка");  
 } catch (NullPointerException e) {  
 System.*out*.println("1");  
 } finally {  
 System.*out*.println("2");  
 }  
 System.*out*.println("3");  
 }  
  
}

Код выполняет следующие действия:

* Выводит на экран строку "0".
* Вызывается оператор throw, который генерирует исключение типа NullPointerException с сообщением "ошибка".
* В блоке catch исключение NullPointerException перехватывается и выводится на экран сообщение "1".
* В блоке finally выводится на экран сообщение "2".
* Выводится на экран строка "3".

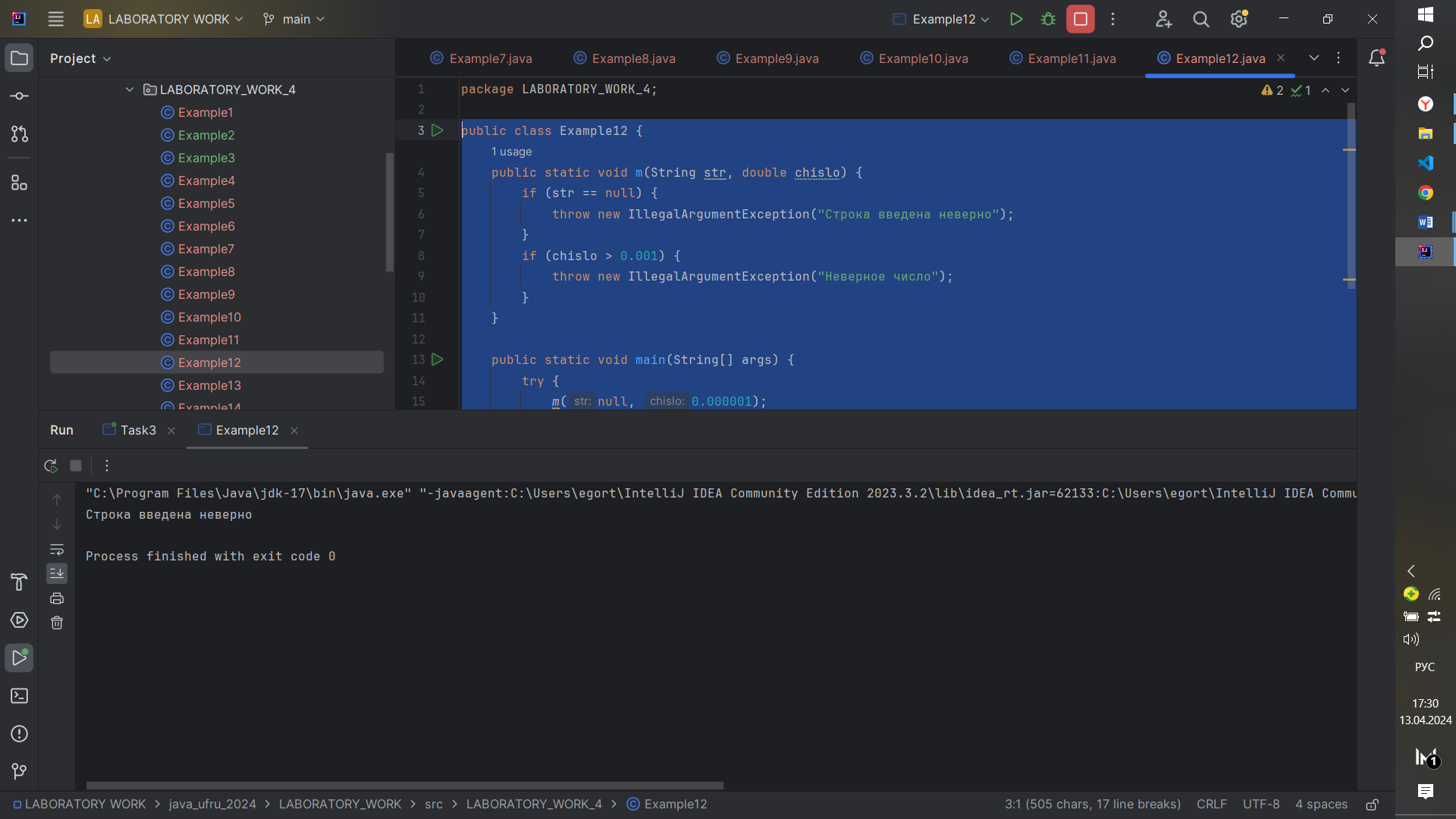


Example12

public class Example12 {  
 public static void m(String str, double chislo) {  
 if (str == null) {  
 throw new IllegalArgumentException("Строка введена неверно");  
 }  
 if (chislo > 0.001) {  
 throw new IllegalArgumentException("Неверное число");  
 }  
 }  
  
 public static void main(String[] args) {  
 try {  
 *m*(null, 0.000001);  
 } catch (IllegalArgumentException e) {  
 System.*out*.println(e.getMessage());  
 }  
 }  
}

Код выполняет следующие действия:

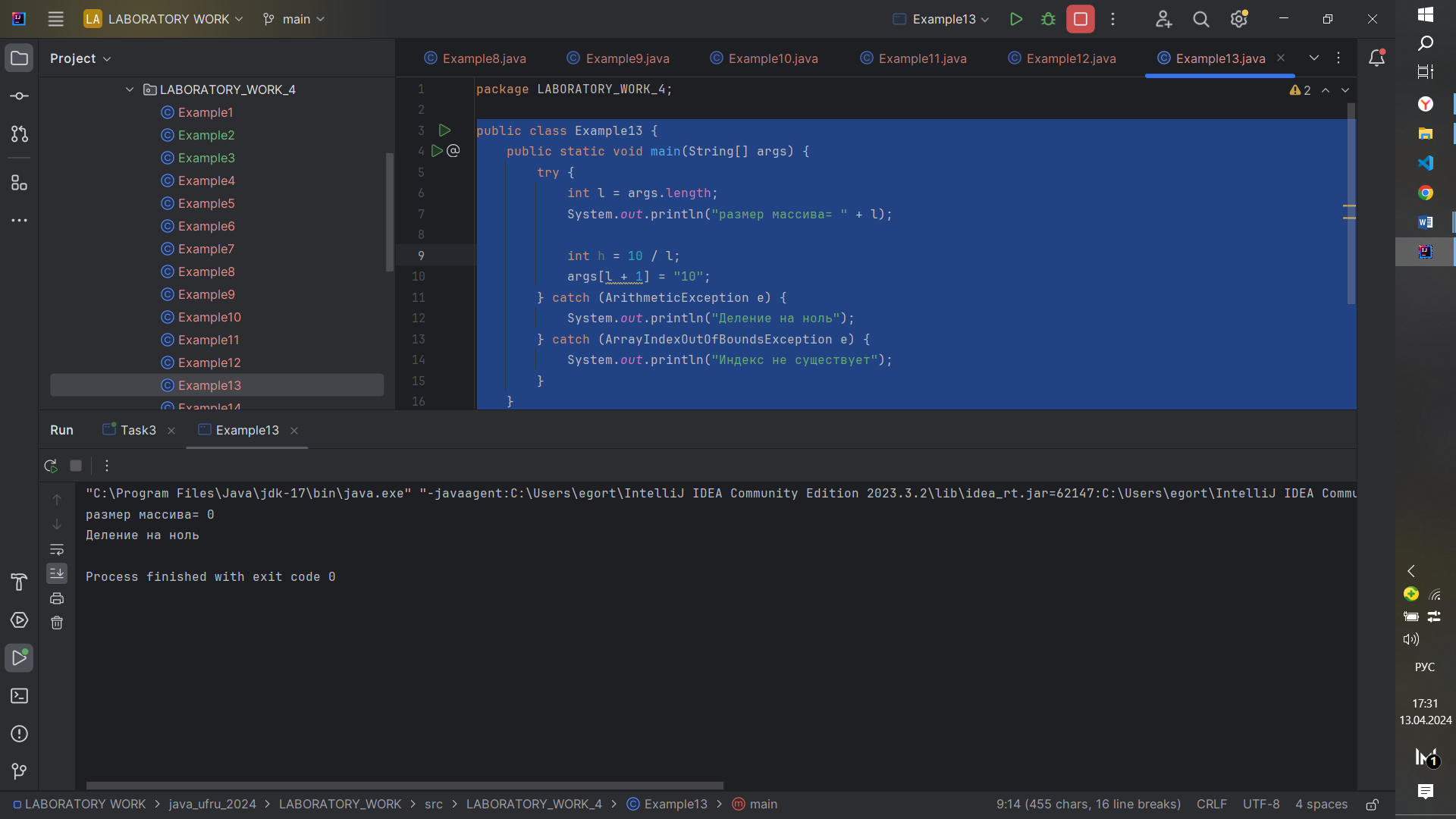
* Вызывается метод m() с аргументами null и 0.000001.
* В блоке if проверяется, является ли аргумент str null. Так как str равен null, генерируется исключение типа IllegalArgumentException с сообщением "Строка введена неверно".
* В блоке catch исключение IllegalArgumentException перехватывается и выводится на экран сообщение "Строка введена неверно".



public class Example13 {  
 public static void main(String[] args) {  
 try {  
 int l = args.length;  
 System.*out*.println("размер массива= " + l);  
  
 int h = 10 / l;  
 args[l + 1] = "10";  
 } catch (ArithmeticException e) {  
 System.*out*.println("Деление на ноль");  
 } catch (ArrayIndexOutOfBoundsException e) {  
 System.*out*.println("Индекс не существует");  
 }  
 }  
  
}

Код выполняет следующие действия

* Вызывается метод main().
* Вычисляется длина массива args и выводится на экран сообщение "размер массива= " + l.
* Выполняется деление 10 на l. Если l равно 0, генерируется исключение типа ArithmeticException.
* Выполняется попытка обращения к элементу массива args с индексом l + 1. Если l больше или равен длине массива args, генерируется исключение типа ArrayIndexOutOfBoundsException.
* В блоке catch исключение ArithmeticException или ArrayIndexOutOfBoundsException перехватывается и выводится на экран соответствующее сообщение.

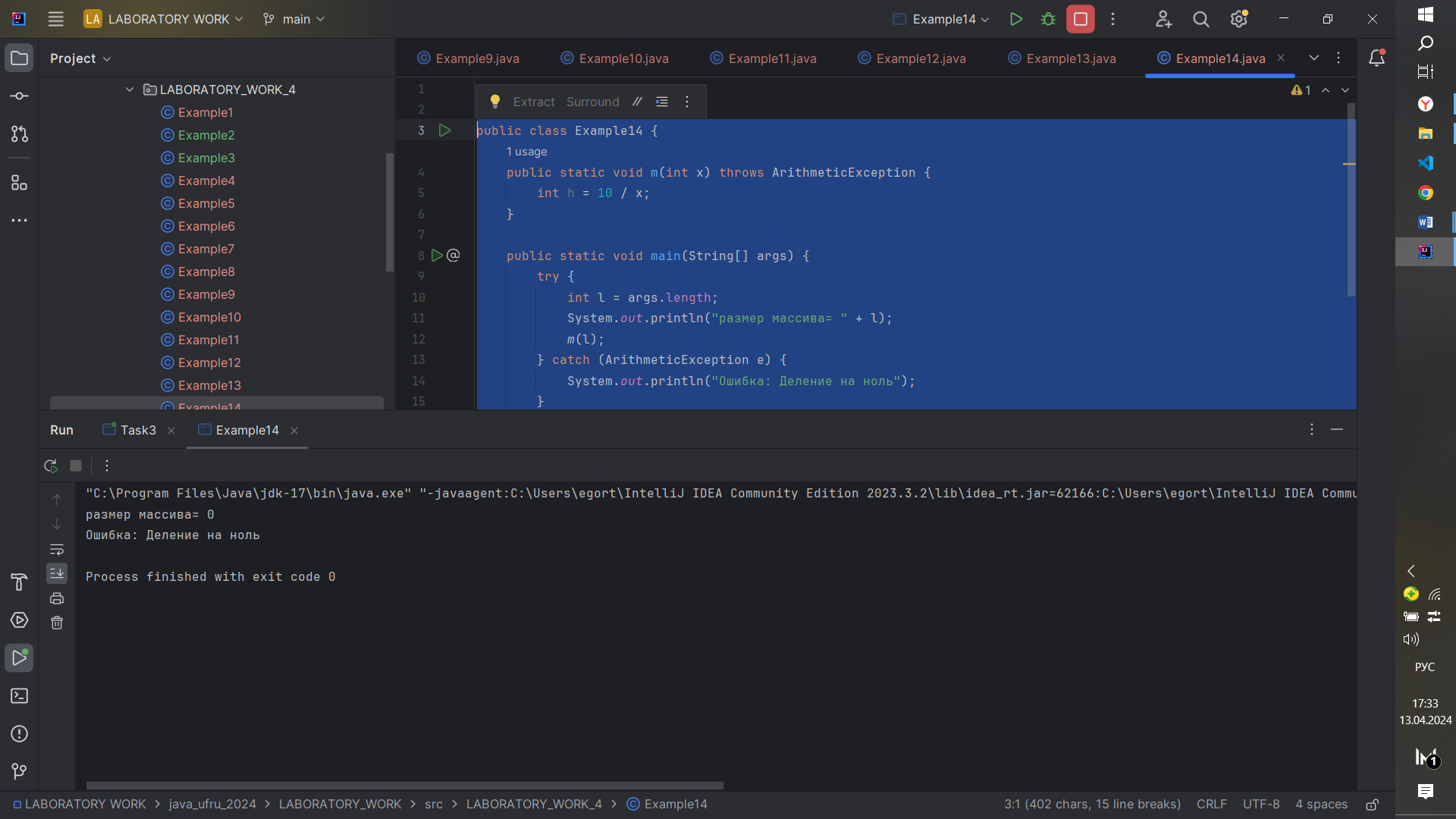


Example14

public class Example14 {  
 public static void m(int x) throws ArithmeticException {  
 int h = 10 / x;  
 }  
  
 public static void main(String[] args) {  
 try {  
 int l = args.length;  
 System.*out*.println("размер массива= " + l);  
 *m*(l);  
 } catch (ArithmeticException e) {  
 System.*out*.println("Ошибка: Деление на ноль");  
 }  
 }  
  
}

Код выполняет следующие действия:

* Вызывается метод main().
* Вычисляется длина массива args и выводится на экран сообщение "размер массива= " + l.
* Вызывается метод m() с аргументом l.
* В методе m() выполняется деление 10 на x. Если x равно 0, генерируется исключение типа ArithmeticException.
* В блоке catch исключение ArithmeticException перехватывается и выводится на экран сообщение "Ошибка: Деление на ноль".

Ч

Task1

public class Task1 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Введите размер массива: ");  
 int size = scanner.nextInt();  
 int[] array = new int[size];  
 int sum = 0;  
 int count = 0;  
  
 System.*out*.println("Введите элементы массива: ");  
 for (int i = 0; i < size; i++) {  
 try {  
 array[i] = scanner.nextInt();  
 if (array[i] > 0) {  
 sum += array[i];  
 count++;  
 }  
 } catch (InputMismatchException e) {  
 System.*out*.println("Ошибка: введены некорректные данные. Повторите ввод.");  
 scanner.next();  
 i--;  
 }  
 }  
  
 if (count == 0) {  
 System.*out*.println("Ошибка: положительные элементы отсутствуют.");  
 } else {  
 double average = (double) sum / count;  
 System.*out*.println("Среднее значение среди положительных элементов массива: " + average);  
 }  
 }  
}

Код выполняет следующие действия:

* Создается объект Scanner для считывания данных с консоли.
* Выводится на экран сообщение "Введите размер массива: " и считывается размер массива.
* Создается массив int[] array размером, введенным пользователем.
* Выводится на экран сообщение "Введите элементы массива: " и считываются элементы массива.
* В цикле for считываются элементы массива и выполняется проверка на положительность. Если элемент положительный, он добавляется в сумму и счетчик.
* Если пользователь вводит некорректные данные (не число), генерируется исключение типа InputMismatchException. В блоке catch исключение перехватывается, выводится на экран сообщение "Ошибка: введены некорректные данные. Повторите ввод." и считывается следующий ввод пользователя.
* Если положительных элементов в массиве нет, выводится на экран сообщение "Ошибка: положительные элементы отсутствуют.".
* Если положительных элементов есть, вычисляется среднее значение среди них и выводится на экран сообщение "Среднее значение среди положительных элементов массива: " + average.

Task2

public class Task2 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Введите размер матрицы: ");  
 int size = scanner.nextInt();  
 int[][] matrix = new int[size][size];  
  
 System.*out*.println("Введите элементы матрицы: ");  
 for (int i = 0; i < size; i++) {  
 for (int j = 0; j < size; j++) {  
 matrix[i][j] = scanner.nextInt();  
 }  
 }  
  
 System.*out*.println("Введите номер столбца, который нужно вывести: ");  
 int columnNumber = 0;  
 try {  
 columnNumber = scanner.nextInt();  
 if (columnNumber < 0 || columnNumber >= size) {  
 throw new IllegalArgumentException("Ошибка: столбец с таким номером отсутствует.");  
 }  
 } catch (InputMismatchException e) {  
 System.*out*.println("Ошибка: введены некорректные данные. Повторите ввод.");  
 scanner.next();  
 } catch (IllegalArgumentException e) {  
 System.*out*.println(e.getMessage());  
 }  
  
 System.*out*.println("Элементы столбца " + columnNumber + ":");  
 for (int i = 0; i < size; i++) {  
 System.*out*.print(matrix[i][columnNumber] + " ");  
 }  
 }  
}

Код выполняет следующие действия:

* Создается объект Scanner для считывания данных с консоли.
* Выводится на экран сообщение "Введите размер матрицы: " и считывается размер матрицы.
* Создается матрица int[][] matrix размером, введенным пользователем.
* Выводится на экран сообщение "Введите элементы матрицы: " и считываются элементы матрицы.
* Выводится на экран сообщение "Введите номер столбца, который нужно вывести: " и считывается номер столбца.
* Если пользователь вводит некорректные данные (не число), генерируется исключение типа InputMismatchException. В блоке catch исключение перехватывается, выводится на экран сообщение "Ошибка: введены некорректные данные. Повторите ввод." и считывается следующий ввод пользователя.
* Если номер столбца меньше 0 или больше размера матрицы, генерируется исключение типа IllegalArgumentException. В блоке catch исключение перехватывается, выводится на экран сообщение "Ошибка: столбец с таким номером отсутствует.".
* Выводится на экран сообщение "Элементы столбца " + columnNumber + ":".
* Выводятся элементы выбранного столбца матрицы.

Task3

public class Task3 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Введите размер массива: ");  
 int size = scanner.nextInt();  
 byte[] array = new byte[size];  
 int sum = 0;  
  
 System.*out*.println("Введите элементы массива: ");  
 for (int i = 0; i < size; i++) {  
 try {  
 array[i] = scanner.nextByte();  
 sum += array[i];  
 } catch (InputMismatchException e) {  
 System.*out*.println("Ошибка: введены некорректные данные. Повторите ввод.");  
 scanner.next();  
 i--;  
 } catch (NumberFormatException e) {  
 System.*out*.println("Ошибка: введенное значение за пределами диапазона типа byte.");  
 scanner.next();  
 i--;  
 }  
 }  
  
 System.*out*.println("Сумма элементов массива: " + sum);  
 }  
}

Код выполняет следующие действия:

Создается объект Scanner для считывания данных с консоли.

Выводится на экран сообщение "Введите размер массива: " и считывается размер массива.

Создается массив byte[] array размером, введенным пользователем.

Выводится на экран сообщение "Введите элементы массива: " и считываются элементы массива.

В цикле for считываются элементы массива и выполняется их суммирование.

Если пользователь вводит некорректные данные (не число), генерируется исключение типа InputMismatchException. В блоке catch исключение перехватывается, выводится на экран сообщение "Ошибка: введены некорректные данные. Повторите ввод." и считывается следующий ввод пользователя.

Если пользователь вводит число, которое не может быть преобразовано в тип byte, генерируется исключение типа NumberFormatException. В блоке catch исключение перехватывается, выводится на экран сообщение "Ошибка: введенное значение за пределами диапазона типа byte." и считывается следующий ввод пользователя.

Выводится на экран сообщение "Сумма элементов массива: " + sum.