**Министерство науки и высшего образования 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

Центр ускоренного обучения

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №13**

**По дисциплине «Программирование»**

Студент группы РИЗ-140938у Бабенко М.А.

Преподаватель: Архипов Н.А.

**Екатеринбург, 2025 г.**

**Лабораторная работа N13**

**«Наследование. Обработка исключительных ситуаций»**

Цель: получение навыков работы с каталогами и файлами операционной системы, а также с классами ввода/вывода, получение навыков ввода/вывода данных файла через символьные потоки.

**Задания для самостоятельной работы**

**Задание 1**. Воспроизвести примеры 1-14 лабораторной работы и при отправке в удаленный репозиторий исправить ошибки.

В главном методе имеется конструкция try-catch, в ней сначала выводится ноль, а потом искуственно выбрасывается исключение типа RuntimeException и выводится сообщение «Непроверяемая ошибка», прерывается выполнение try и выполняется catch.

Результат работы программы и ее код показан на рисунке 1.

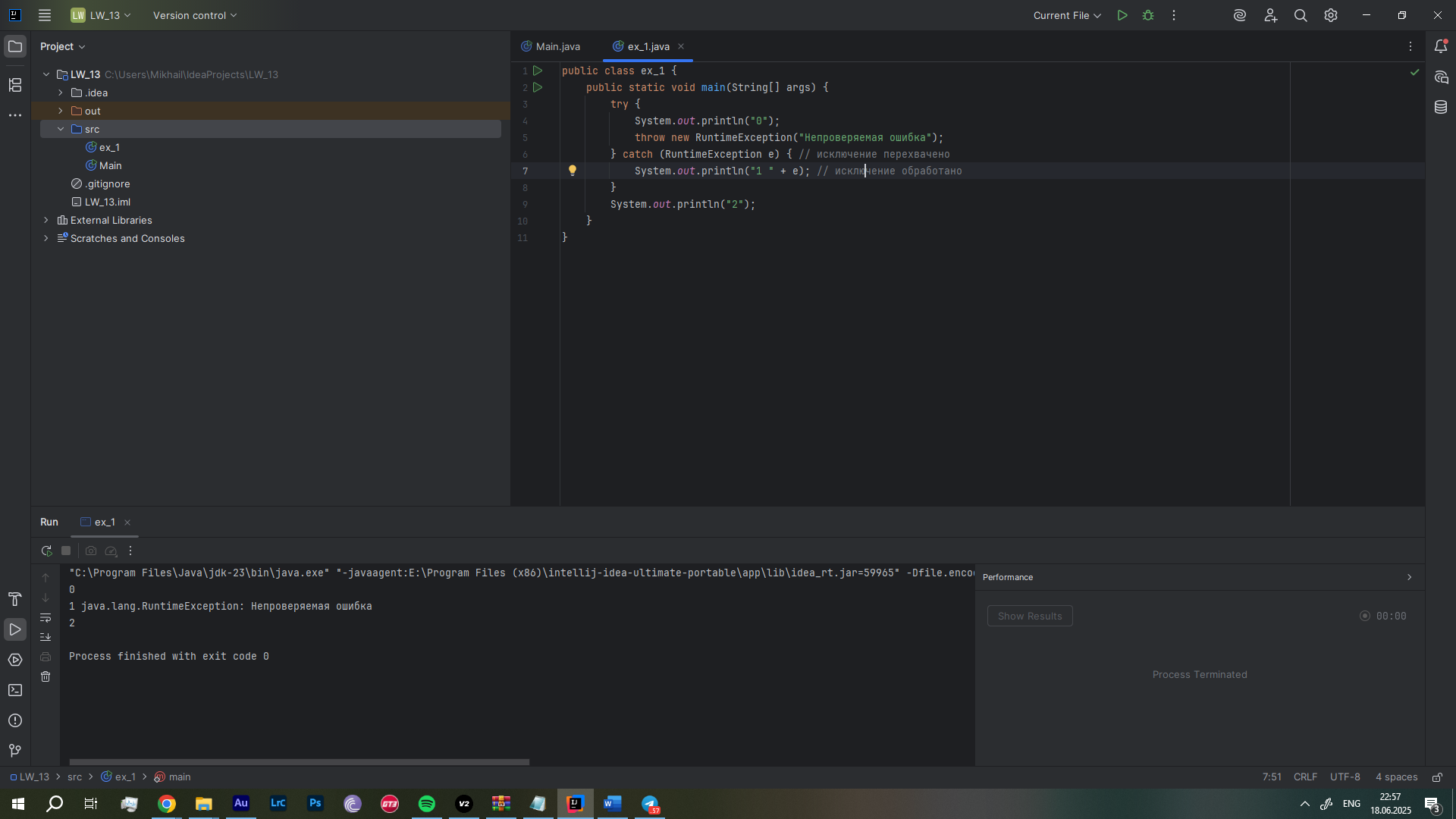


Рисунок 1 – Код и результат работы программы ex\_1

В главном методе имеется конструкция try-catch, в ней сначала выводится ноль, а потом искуственно выбрасывается исключение типа RuntimeException и выводится сообщение «Непроверяемая ошибка», прерывается выполнение try и выполняется catch. В то время когда выполняется catch блок, выводится 2, а после его выплнения, выводится 3. Исправленная ошибка в коде – команда println для единицы была недостижима, т.к. находилась пос выброшенной ошибки.

Результат работы программы и ее код показан на рисунке 2.

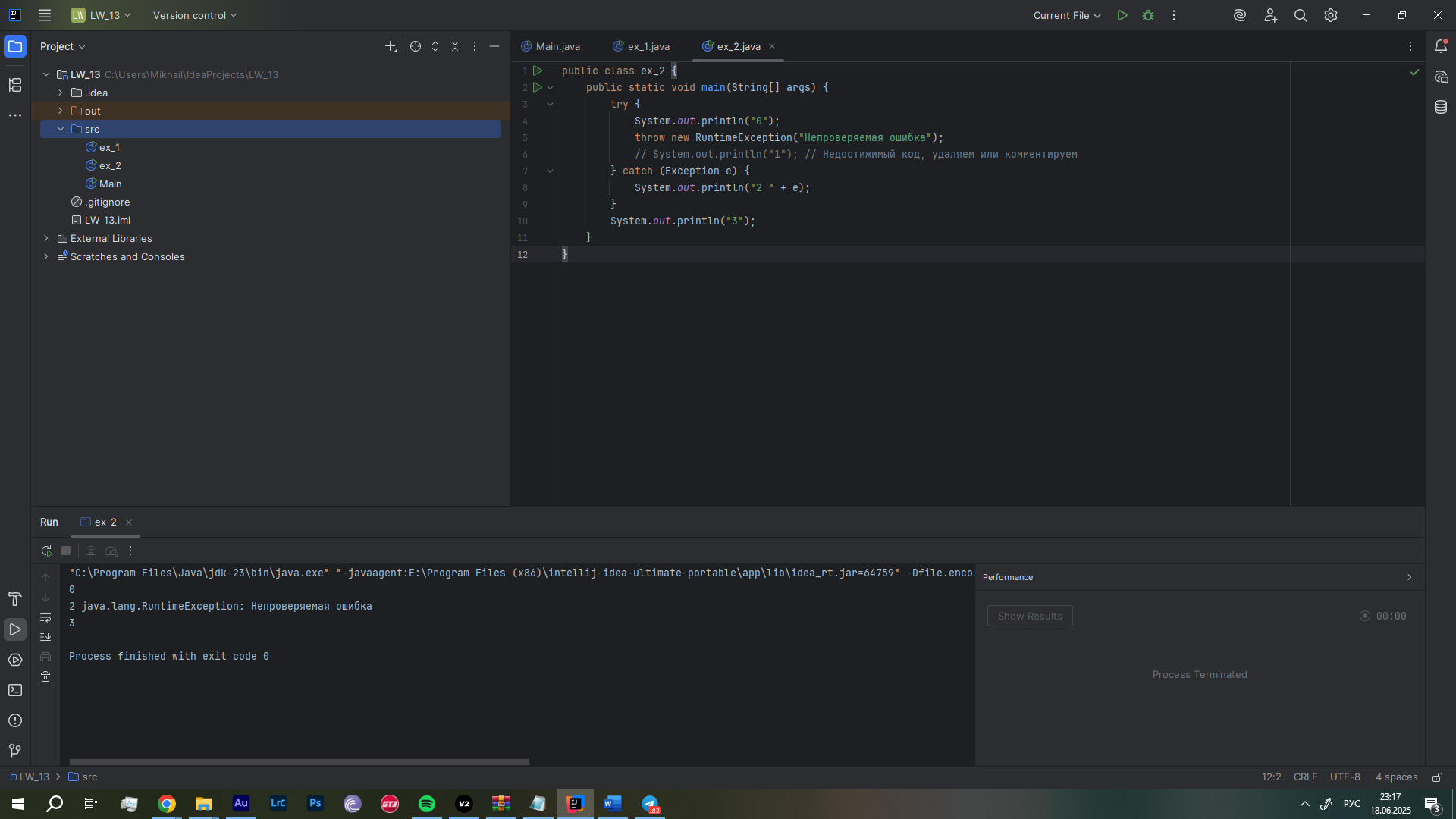


Рисунок 2 – Код и результат работы программы ex\_2

В главном методе имеется конструкция try-catch, в ней сначала выводится ноль, а потом искуственно выбрасывается исключение типа RuntimeException и выводится сообщение «Непроверяемая ошибка», прерывается выполнение try и выполняется catch. Далее, в зависимости от ошибки выполняется разный блок catch – в нашем случае выполняется для исключения RuntimeException e, будет в консоль вывоедено число 2, а потом и 4 (после всех catch-блоков).

Результат работы программы и ее код показан на рисунке 3.

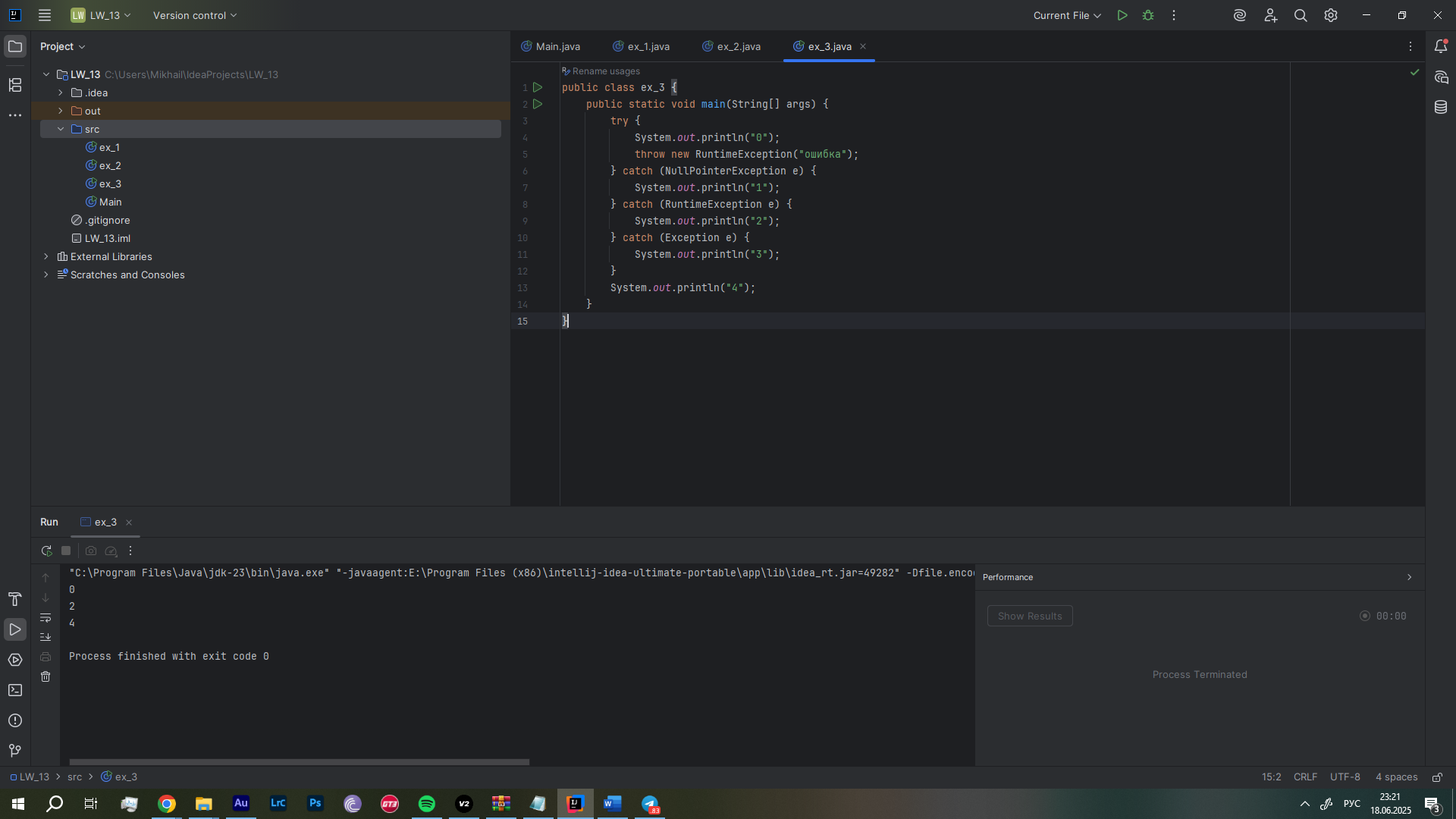


Рисунок 3 – Код и результат работы программы ex\_3

В главном методе имеется конструкция try-catch, в ней сначала выводится ноль, а потом искуственно выбрасывается исключение типа RuntimeException и выводится сообщение «Непроверяемая ошибка», прерывается выполнение try и выполняется catch. Далее, в зависимости от ошибки выполняется разный блок catch – в нашем случае выполняется для исключения RuntimeException e, будет в консоль вывоедено число 2, а потом и 4 (после всех catch-блоков).

Результат работы программы и ее код показан на рисунке 4.

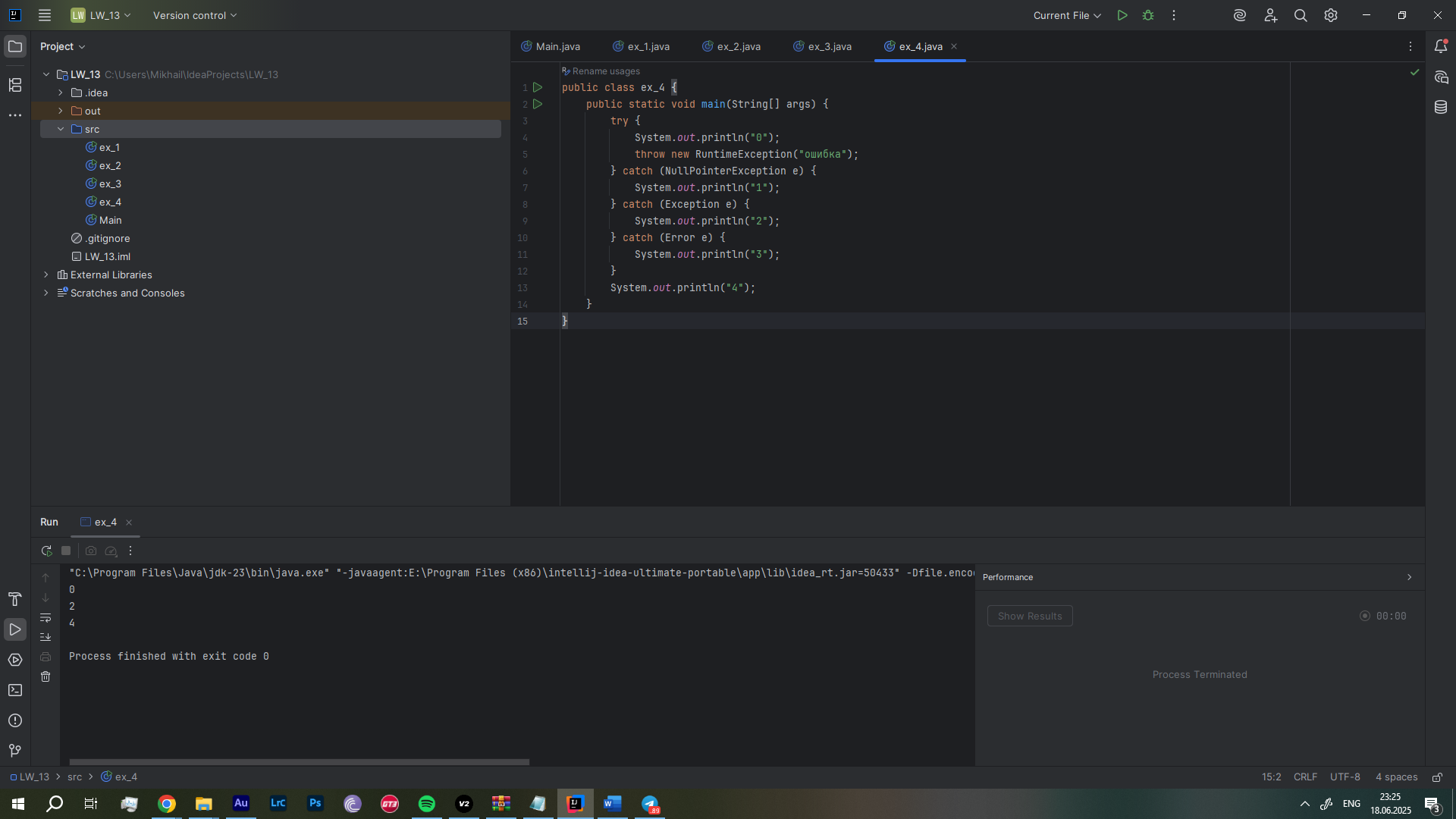


Рисунок 4 – Код и результат работы программы ex\_4

В главном методе имеется конструкция try-catch, в ней сначала выводится ноль, а потом искуственно выбрасывается исключение типа RuntimeException и выводится сообщение «Непроверяемая ошибка», прерывается выполнение try и выполняется catch. Далее, присутствует блок catch – в нашем случае не выполняется, т.к. он предназначен для исключения NullPointerException e, будет в консоль выведено число 0, а сообщение о том, что программа завершена с ошибками. Для исправления этого добавляем блок catch с ошибкой, которая искуственно выбрасывается – RuntimeException e.

Результат работы программы и ее код показан на рисунке 5.

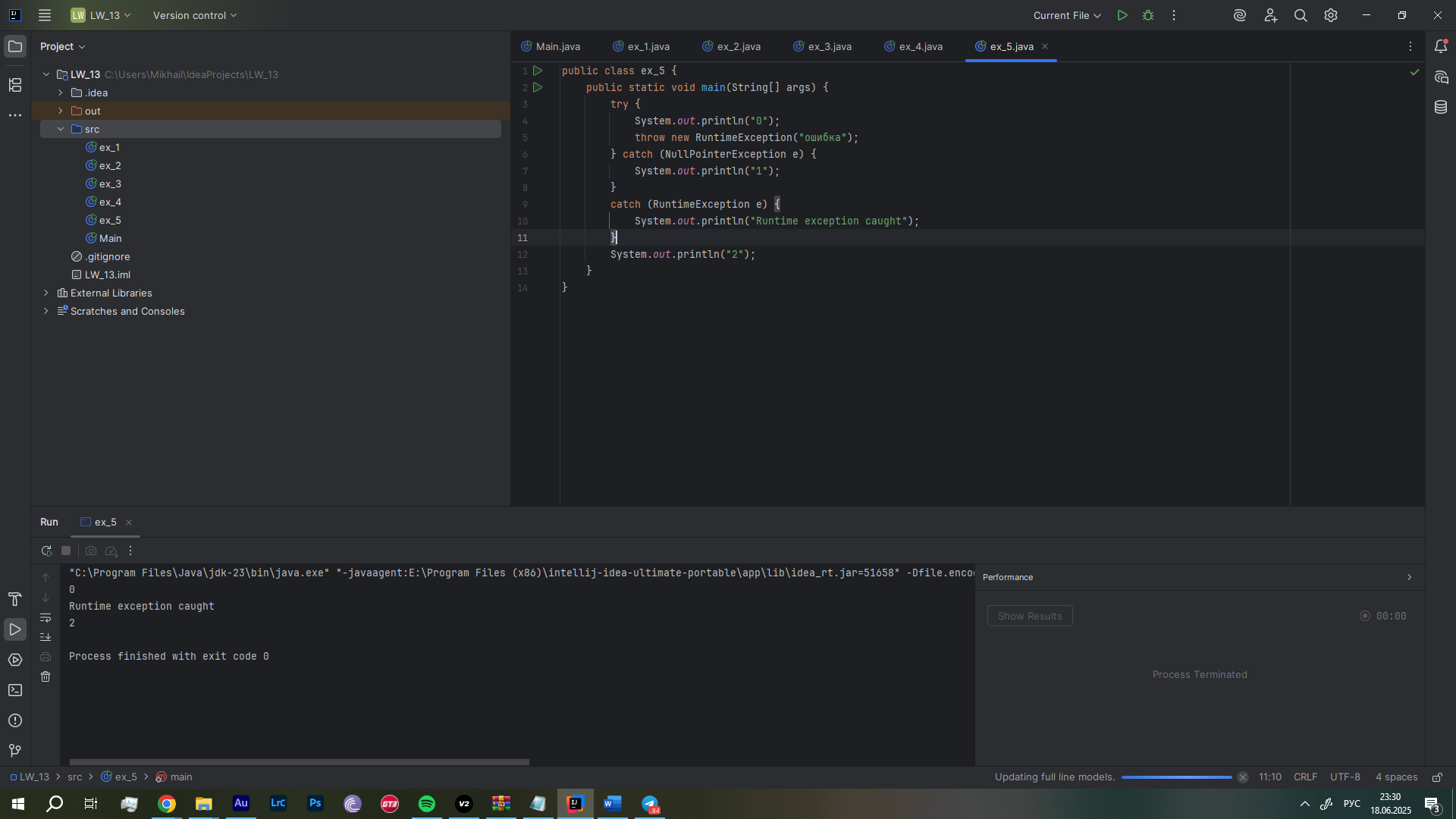


Рисунок 5 – Код и результат работы программы ex\_5

В главном методе имеется конструкция try-catch, в ней сначала выводится ноль, а потом искуственно выбрасывается исключение типа NullPointerException и выводится сообщение «Непроверяемая ошибка», прерывается выполнение try и выполняется catch.

Далее, в зависимости от ошибки выполняется должен выполняться разный блок catch – но в программе сейчас нет нужного. В программе есть ошибка - RuntimeException e стоит раньше, чем Exception. Тогда компиляция будет успешной, будет в консоль выведено число 3, а потом и 4 (после всех catch-блоков).

Результат работы программы и ее код показан на рисунке 6.

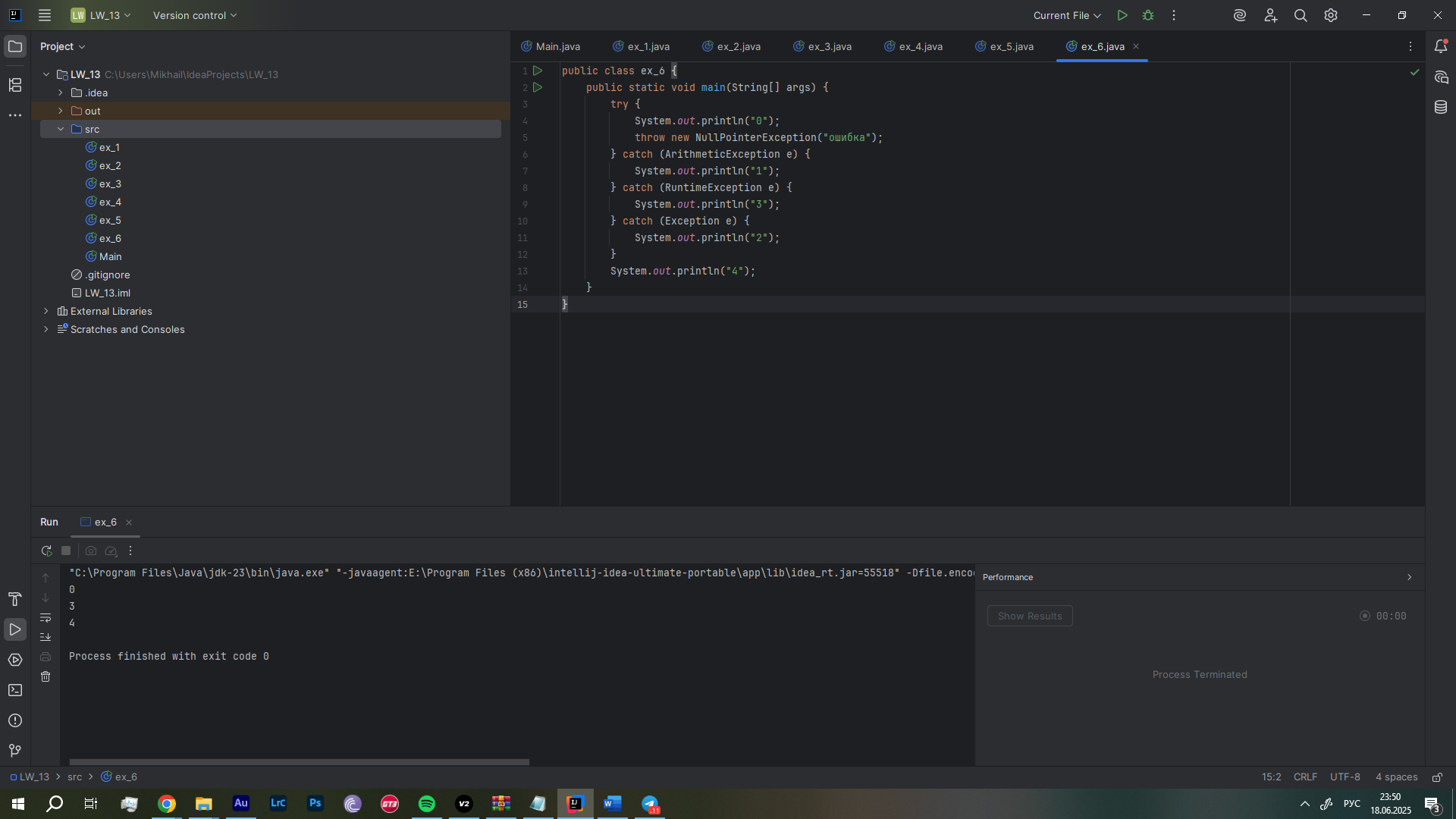


Рисунок 6 – Код и результат работы программы ex\_6

В главном методе имеется конструкция try-catch, в ней сначала выводится ноль, а потом искуственно выбрасывается исключение типа NullPointerException и выводится сообщение «Непроверяемая ошибка», прерывается выполнение try и выполняется catch. Далее, в зависимости от ошибки выполняется разный блок catch – в нашем случае выполняется для исключения NullPointerException e, будет в консоль выведено число 1, а затем будет выброшена ошибка ArithmeticException, компиляция завершается с ошибкой. Чтобы ее исправить, нужно обернуть throw внутри try-catch, тогда второе исключение будет успешно выброшено, в консоль выведется 3, а потом и 4 (после всех catch-блоков).

Результат работы программы и ее код показан на рисунке 7.

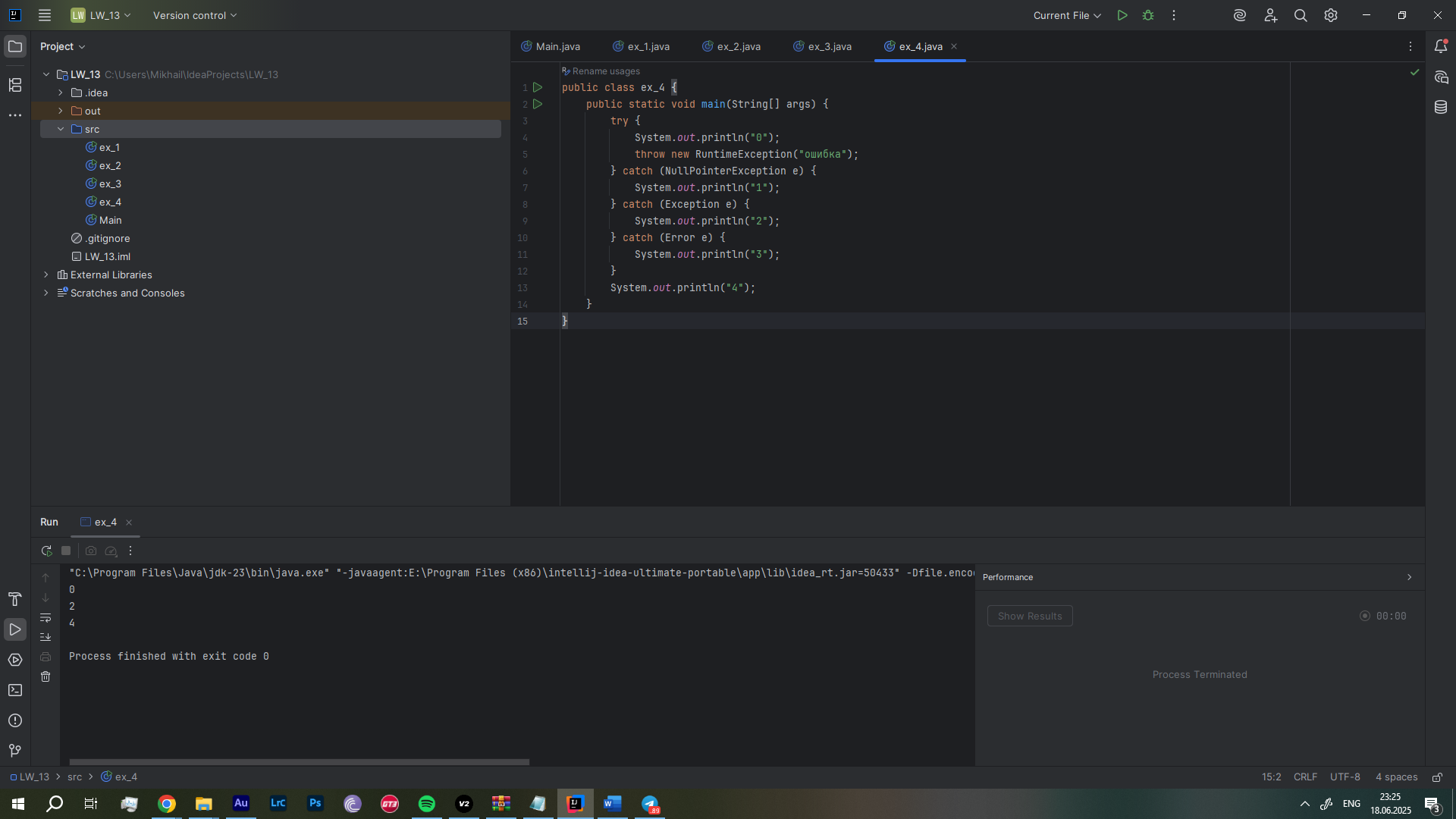


Рисунок 7 – Код и результат работы программы ex\_7

В методе m имеется конструкция try-catch, в ней сначала выводится ноль, а потом искуственно выбрасывается исключение типа RuntimeException, выполняется finally и выводится единица, а затем возникает ошибка компиляции. Чтобы ее исправить, нужно в главный метод добавить конструкцию try-catch, которое будет обрабатывать ошибку, тогда исключение будет успешно выброшено, в консоль выведется сообщение об ошибке.

Результат работы программы и ее код показан на рисунке 8.

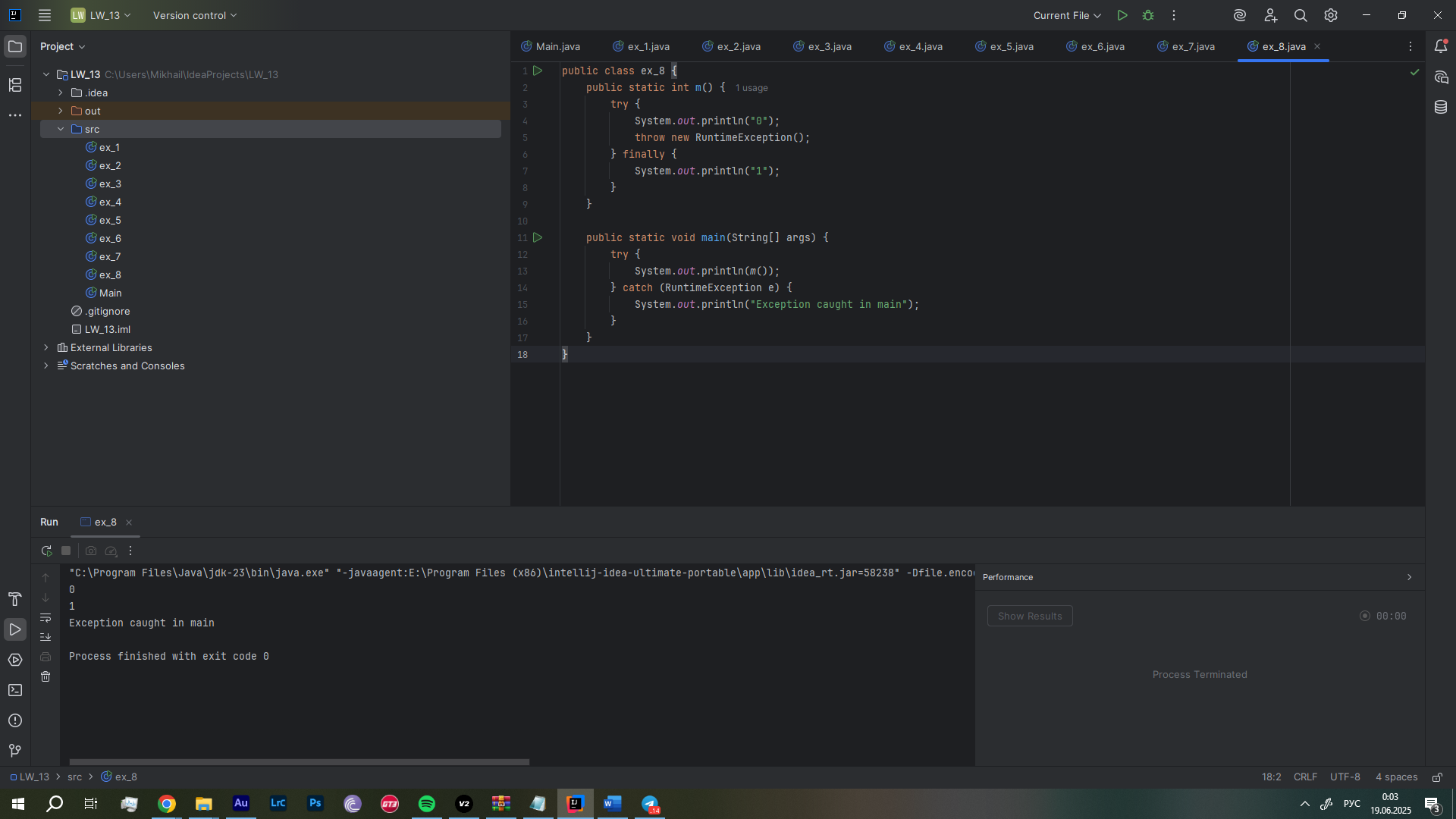


Рисунок 8 – Код и результат работы программы ex\_8

В методе m имеется конструкция try-finally, в ней сначала выводится ноль, а потом выполняется finally и выводится единица, по выполнении метода возвращается значение 55. В главном методе это значение будет выведено путем успешного выполнения метода. Ошибок в коде нет, единственное – finally всегда выполняется, даже если в try есть return.

Результат работы программы и ее код показан на рисунке 9.

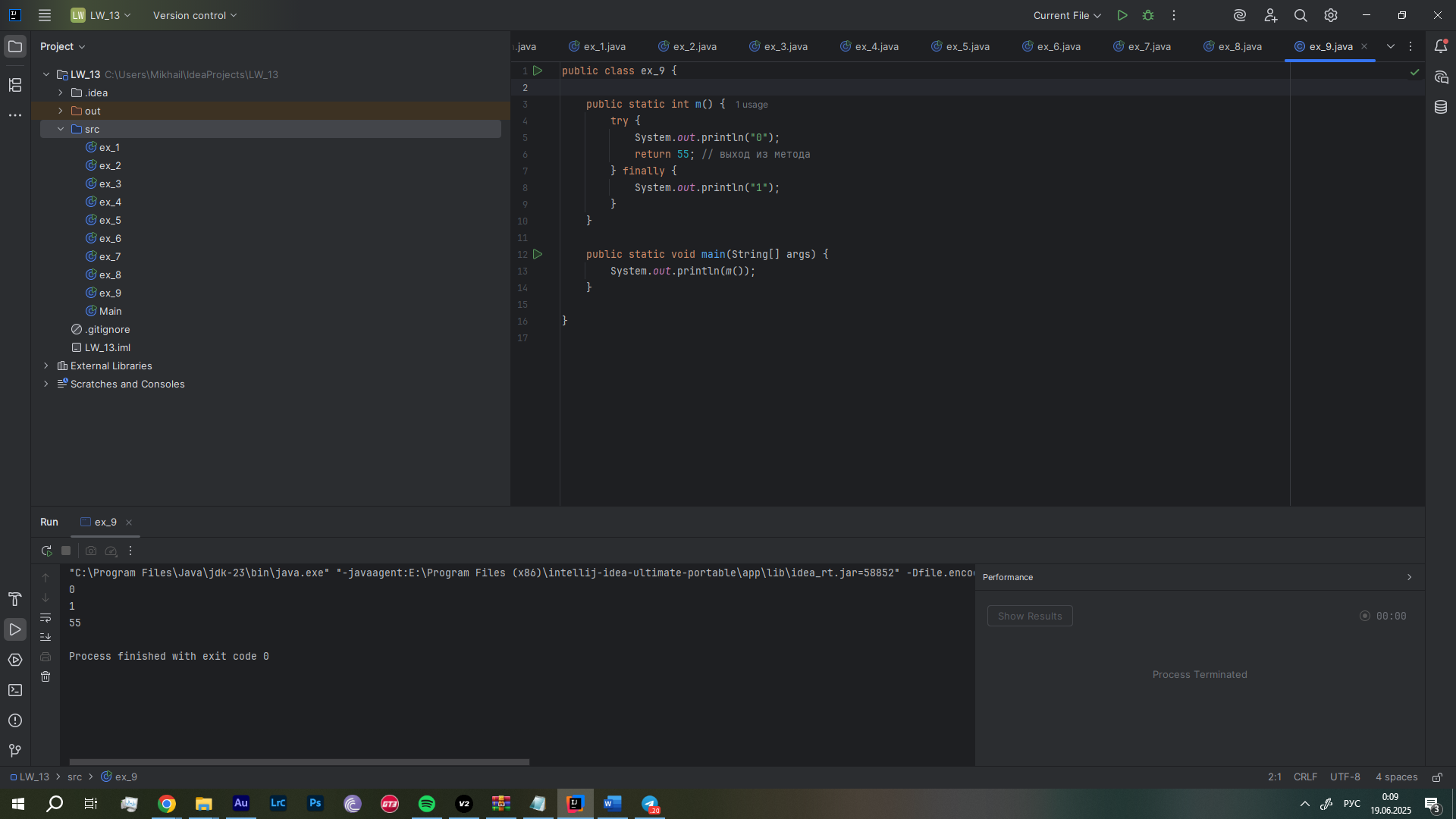


Рисунок 9 – Код и результат работы программы ex\_9

В методе m имеется конструкция try-finally, в ней сначала выводится ноль, а потом выполняется finally и выводится единица, по выполнении метода возвращается значение 20. В главном методе это значение будет выведено путем успешного выполнения метода. Ошибок в коде нет, единственное – программа готова вывести значение 15, но выполнение перехватывается finally, и возвращается 20. Чтобы этого не было, можно убрать return в finally, тогда будет возвращаться 15.

Результат работы программы и ее код показан на рисунке 10.

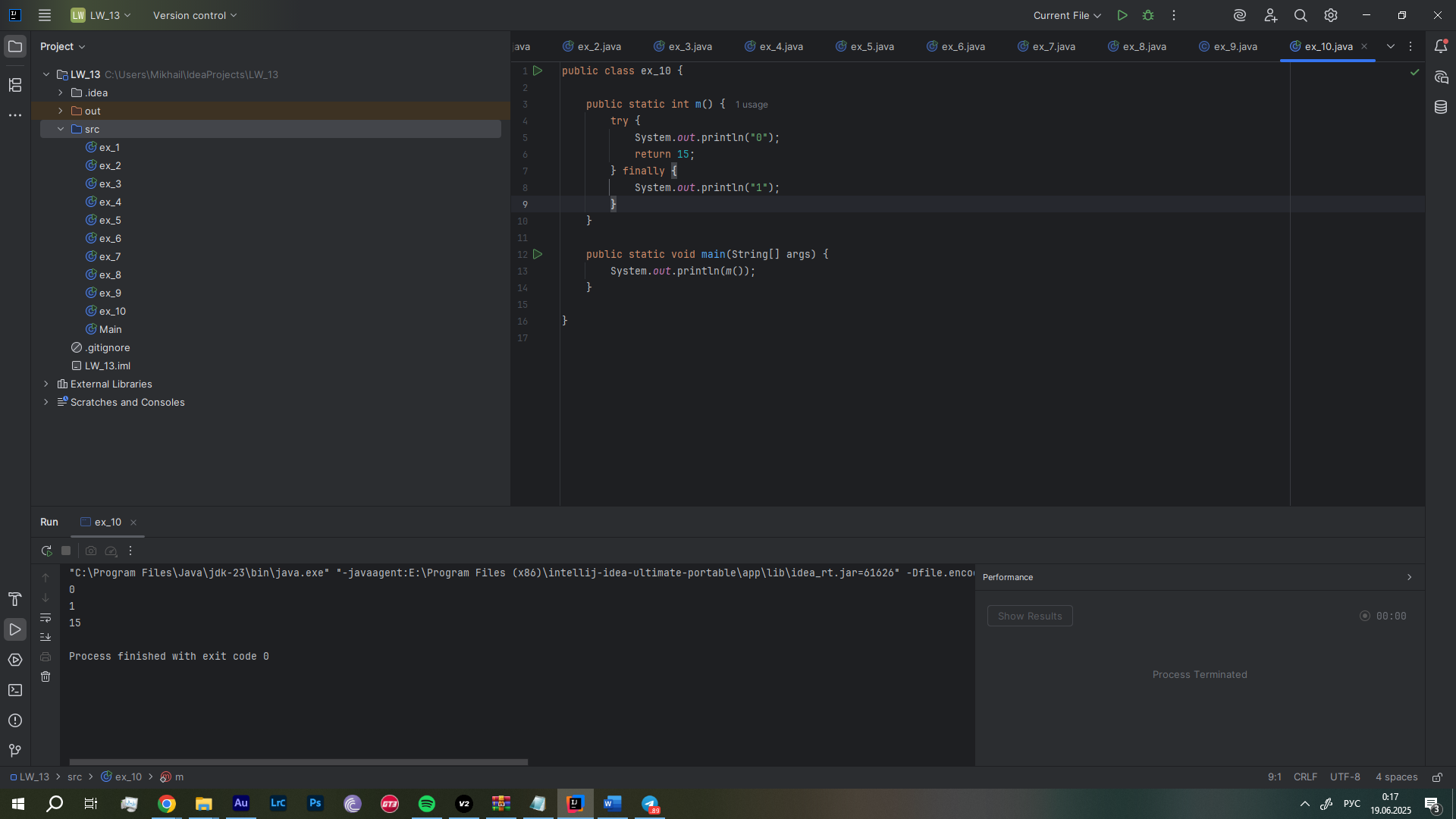


Рисунок 10 – Код и результат работы программы ex\_10

В главном методе имеется конструкция try-catch, в ней сначала выводится ноль, а потом искуственно выбрасывается исключение типа NullPointerException и выводится сообщение «Непроверяемая ошибка», прерывается выполнение try и выполняется catch. Далее, в зависимости от ошибки выполняется разный блок catch – в нашем случае выполняется для исключения NullPointerException e, будет в консоль выведено число 1, а затем выполнен блок finally, будет выведено 2, и 3 после выполнения всей этой конструкции. Компиляция прошла без ошибок.

Результат работы программы и ее код показан на рисунке 11.

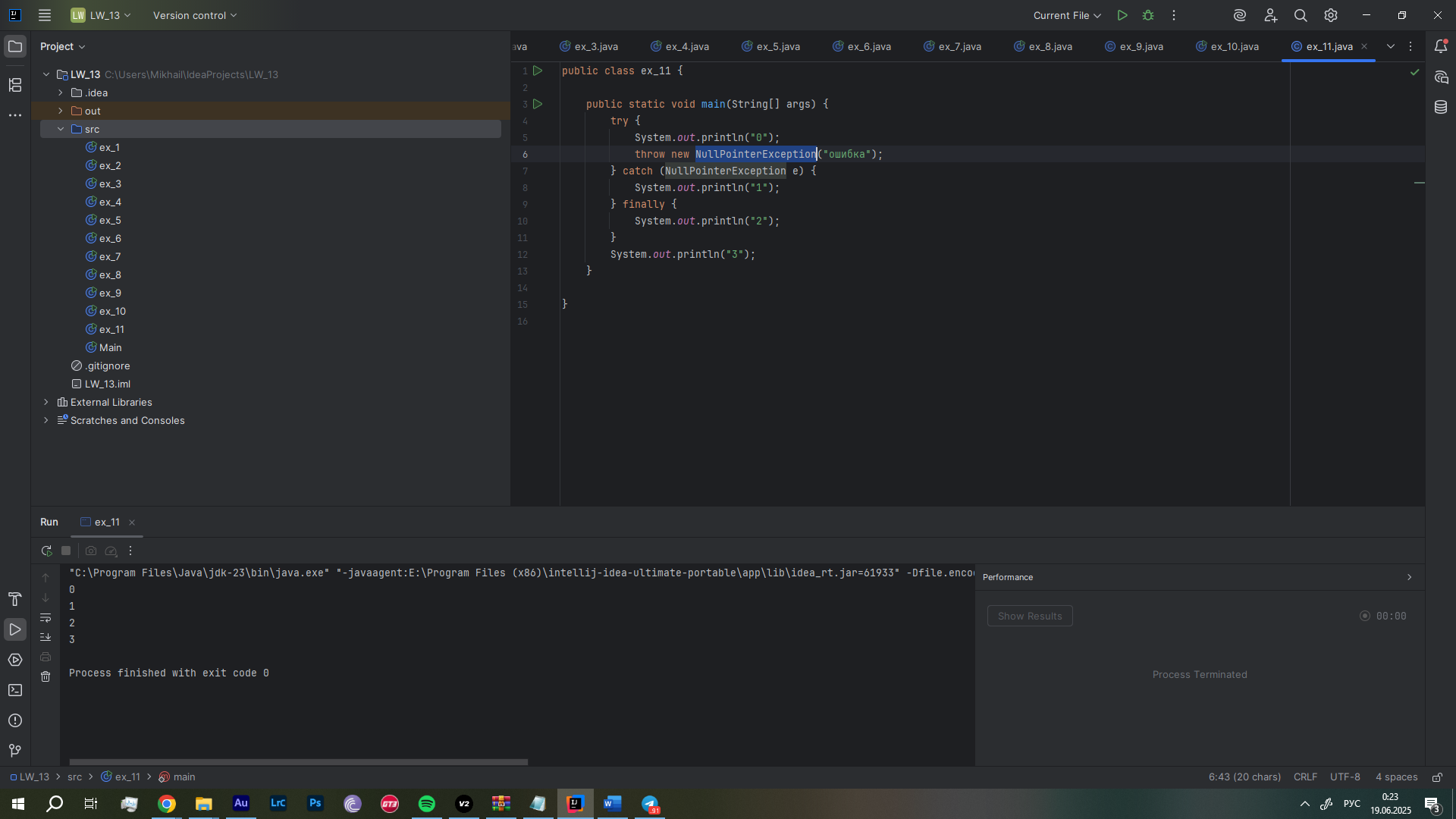


Рисунок 11 – Код и результат работы программы ex\_11

В методе m имеются условия if, если число меньше чем 0.001, будет выброшена ошибка IllegalArgumentException и выведено что число неверное. Если str строка пустая, то будет выброшена ошибка IllegalArgumentException и сообщение о том что строка введена неверно. Компиляция кода произошла с ошибкой – нет обработчика исключения, оно выбрасывается, но не обрабатывается. Для того чтобы исправить это, добавляем обработчик исключения IllegalArgumentException e, в котором будет выводиться сообщение об ошибке.

Результат работы программы и ее код показан на рисунке 12.

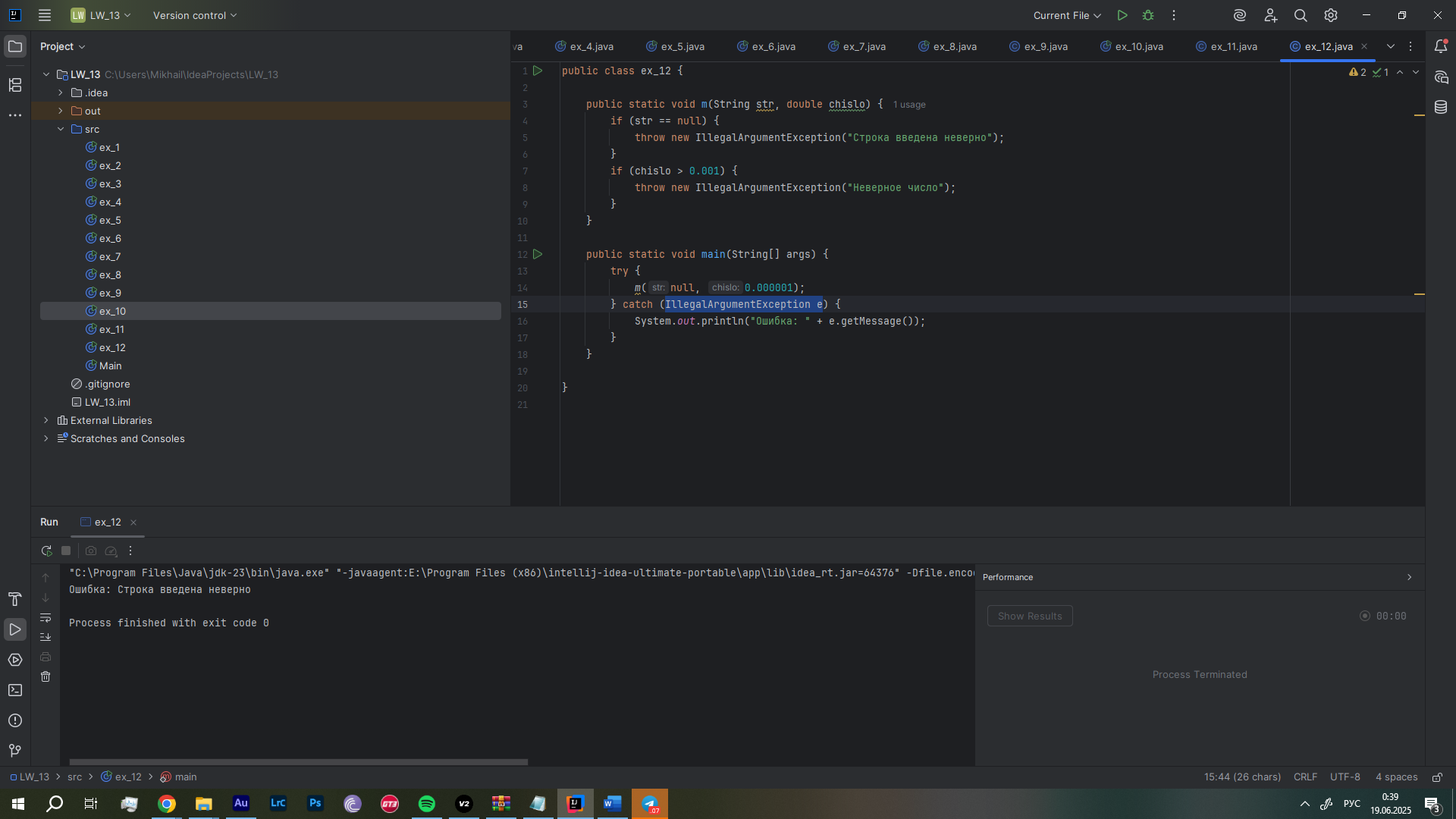


Рисунок 12 – Код и результат работы программы ex\_12

В главном методе имеется конструкция try-catch, в нем выводится размер массива, далее производится попытка деления на 0, и выбрасывается ошибка ArithmeticException e, которая обрабатывается в catch блоке, в нем отображается соответствующее сообщение, программа завершается. В коде нет ошибок по синтаксису, он компилируется и выполняется.

Результат работы программы и ее код показан на рисунке 13.

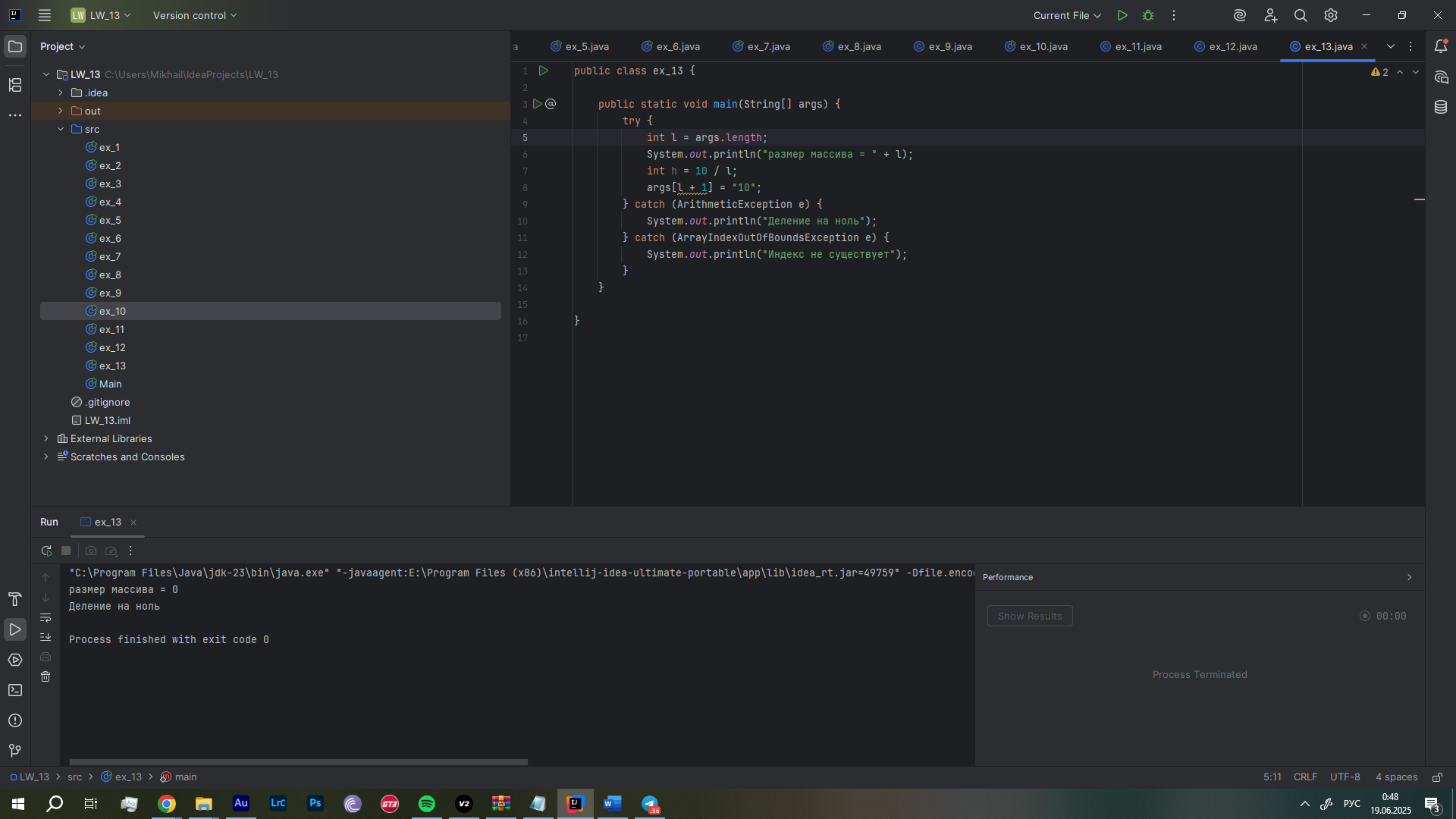


Рисунок 13 – Код и результат работы программы ex\_13

В методе m имеется операция деления на 0, будет выброшена ошибка IllegalArgumentException и выведено что число неверное. Если str строка пустая, то будет выброшена ошибка ArithmeticException, и сообщение о том что произошло деление на 0. В коде нет ошибок по синтаксису, он компилируется и выполняется. Единственное – метод m ничего не делает, и он бесполезен – решено добавить ему return.

Результат работы программы и ее код показан на рисунке 14.

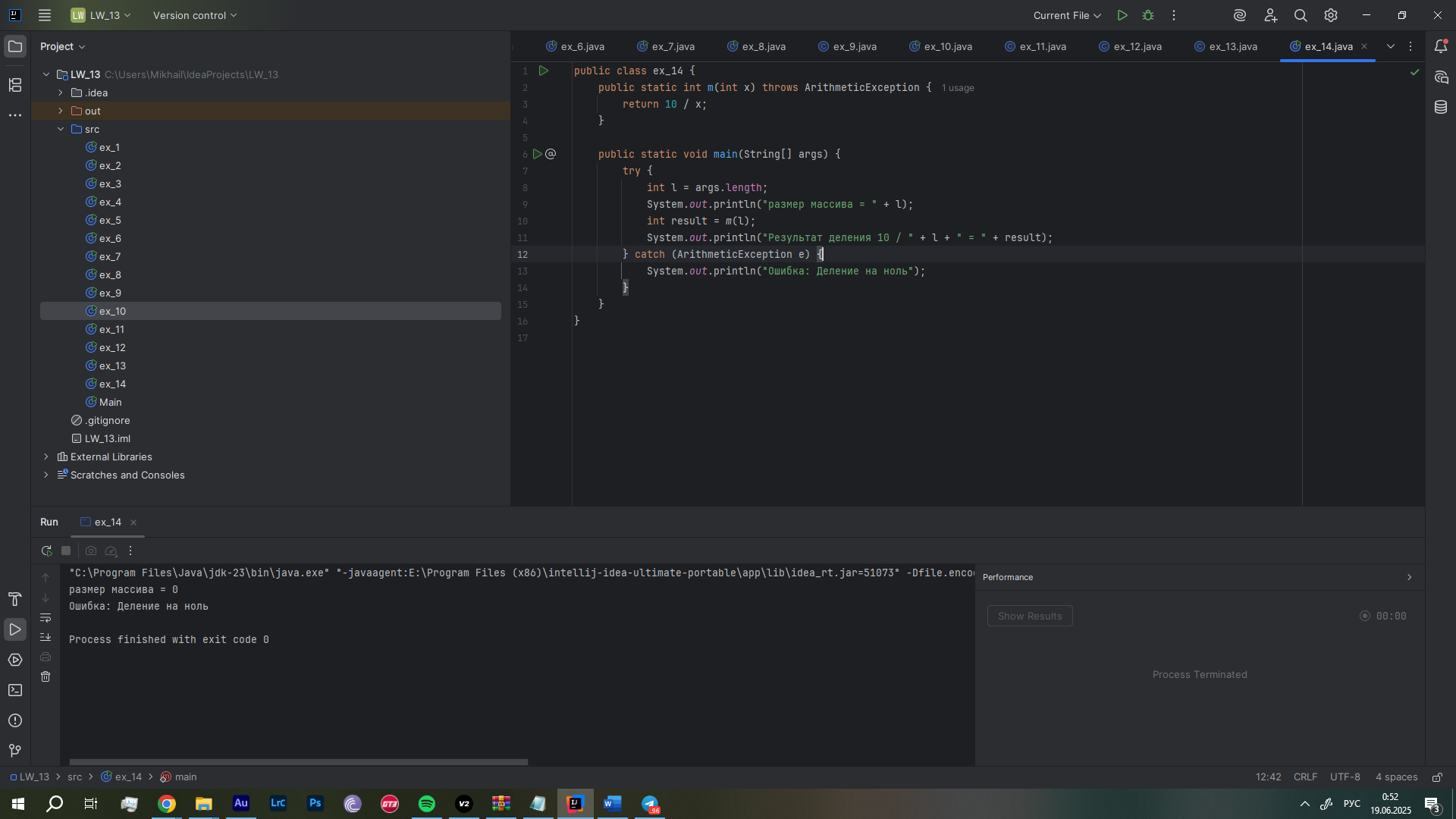


Рисунок 14 – Код и результат работы программы ex\_14

**Вывод**

В результате лабораторной работы были получены навыки работы с каталогами и файлами операционной системы, а также с классами ввода/вывода, получение навыков ввода/вывода данных файла через символьные потоки.