МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

«МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ СВЯЗИ И ИНФОРМАТИКИ»

(МТУСИ)

Кафедра «Математическая Кибернетика и Информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

по дисциплине

«Информационные технологии и программирование»

на тему

“Исключения и их обработка”

Выполнил:

студент группы БВТ2302

Миронов А. А.

Москва, 2024 г.

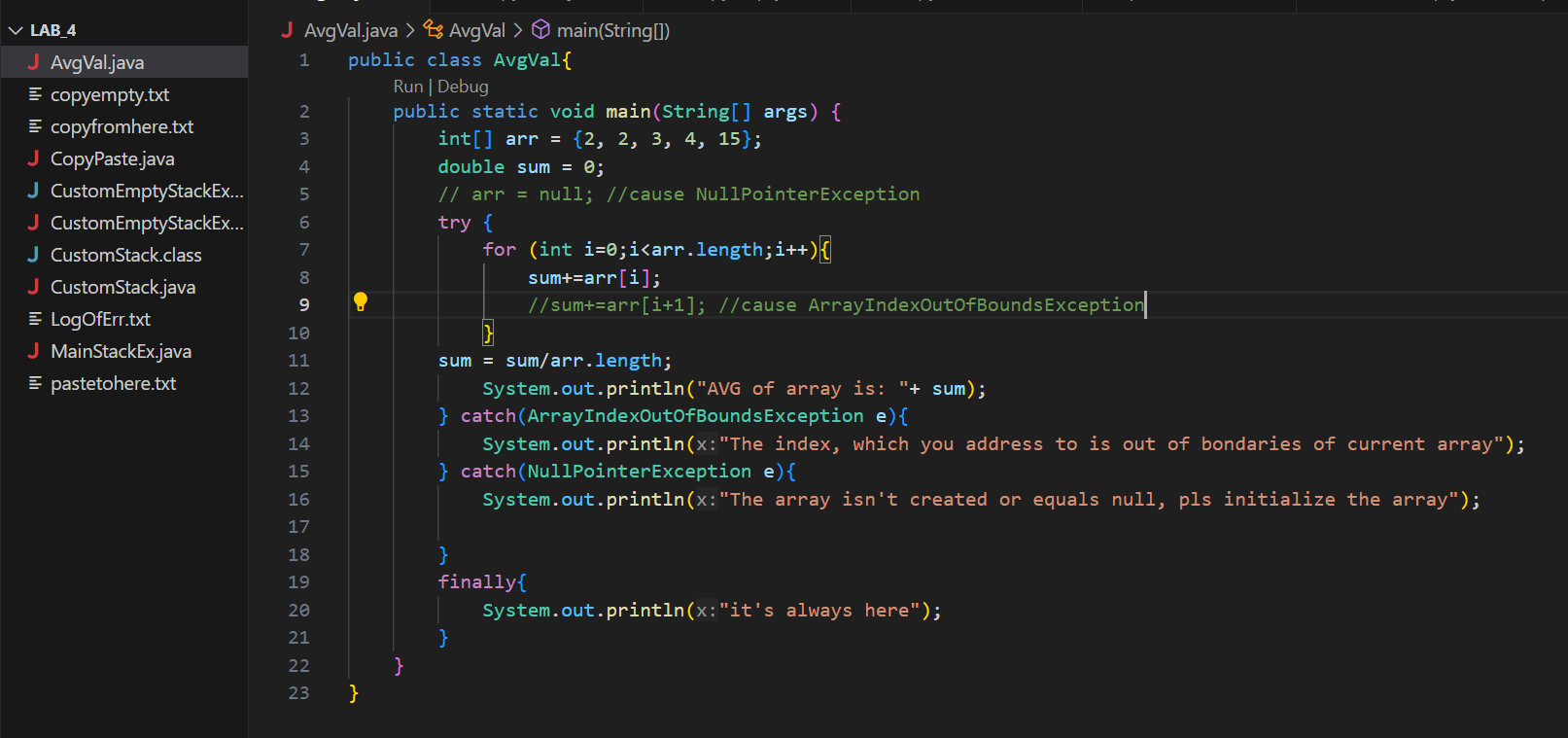
**Задание 1:**

Необходимо написать программу, которая будет находить среднее

арифметическое элементов массива. При этом программа должна

обрабатывать ошибки, связанные с выходом за границы массива и

неверными данными.

****

Создадим метод мэйн, который считает среднее арифметическое аргументов массива. Облачим часть кода, которая может вызывать исключения в блок try, а далее обработаем исключения, при которых мы выходим за границы массива, или мы пытаемся работать с массивом, который является неинициализированным (=null). Блок finally исполняется, даже если было выброшено исключение.

**Задание 2:**

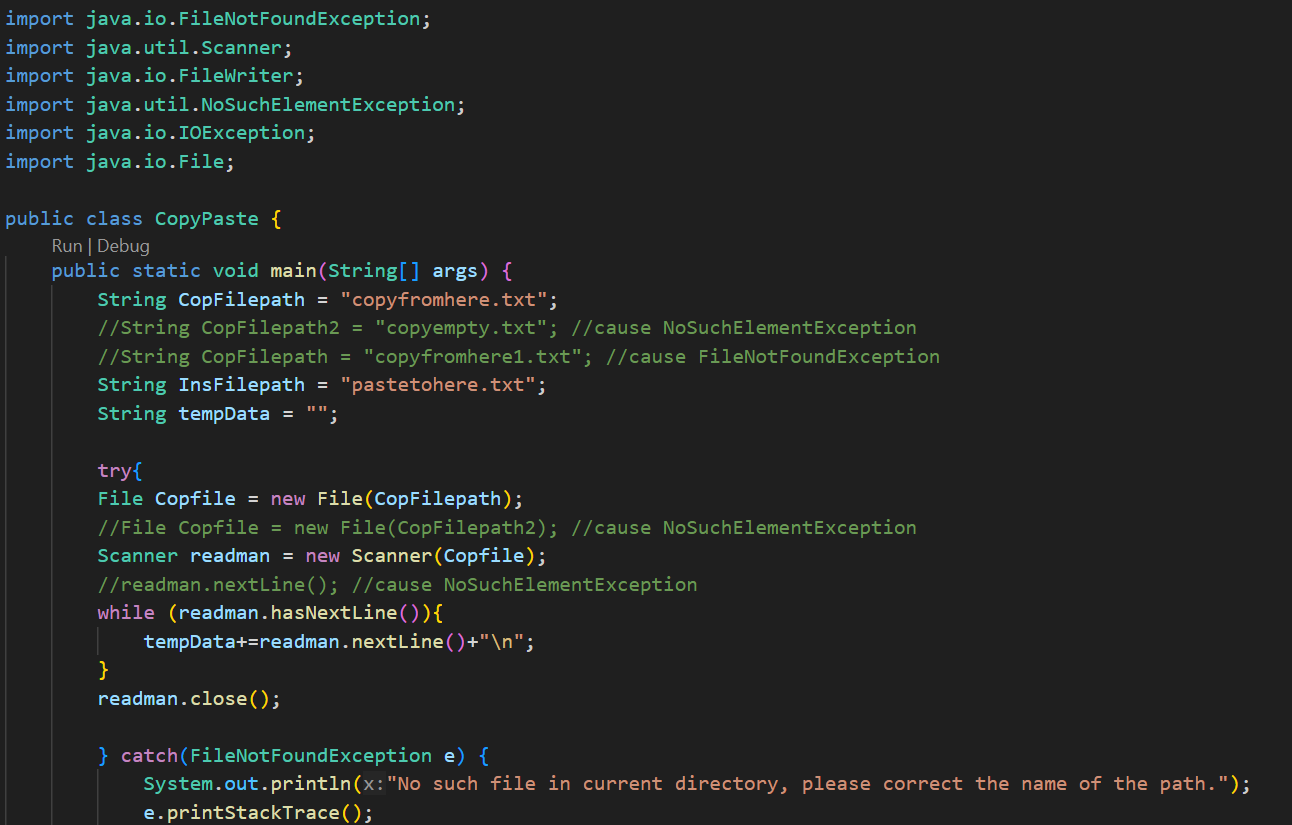
Необходимо написать программу, которая будет копировать содержимое

одного файла в другой. При этом программа должна обрабатывать

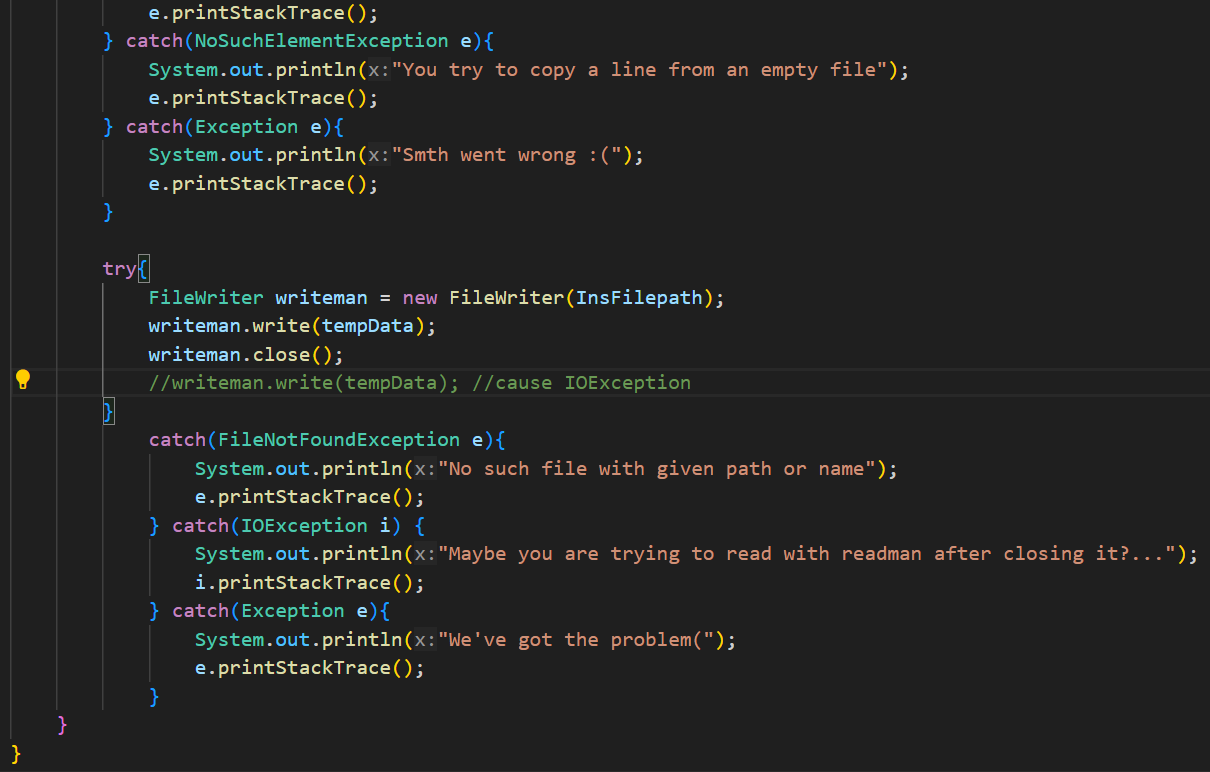
возможные ошибки, связанные с:

Вариант 1 Вариант 2

Открытием и закрытием файлов Чтение и записью файлов

****

Для начала импортируем все необходимые элементы, а это исключения, которые мы будем обрабатывать и классы и для копирования текста из одного файла и записи в другой. FileNotFoundException выбрасывается в случае, когда наш файл не найден по указанному пути. Метод printStackTrace() выводит в консоль обширную информацию о выброшенном исключении.

****

NoSuchElementException появляется, когда мы пытаемся скопировать строку из пустого файла с помощью сканнера. IOException же возникает, когда мы после закрытия врайтера пытаемся записать что то в другой файл. Стоит отметить, xnj в конце конструкции try-catch мы располагаем исключение Exception, которое стоит в классификации исключений выше двух предыдущих и соответствует любому исключению при его обработке. Чем выше иерархия исключения, тем ниже оно стоит в структуре try-catch.

**Задание 3:**

Создайте Java-проект для работы с исключениями. Для каждой из восьми

задач, напишите свой собственный класс для обработки исключений.

Создайте обработчик исключений, который логирует информацию о

каждом выброшенном исключении в текстовый файл.

***Вариант 5:***

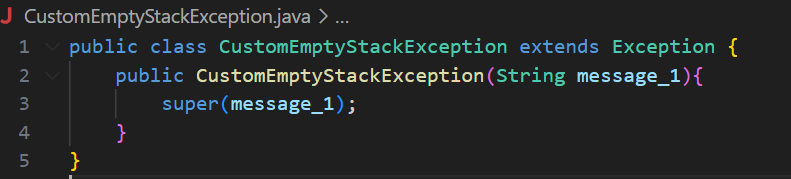
Создайте класс CustomEmptyStackException, который будет

использоваться для обработки исключения EmptyStackException.

Напишите класс CustomStack, имитирующий стек, и, если происходит

попытка извлечь элемент из пустого стека, выбрасывайте исключение

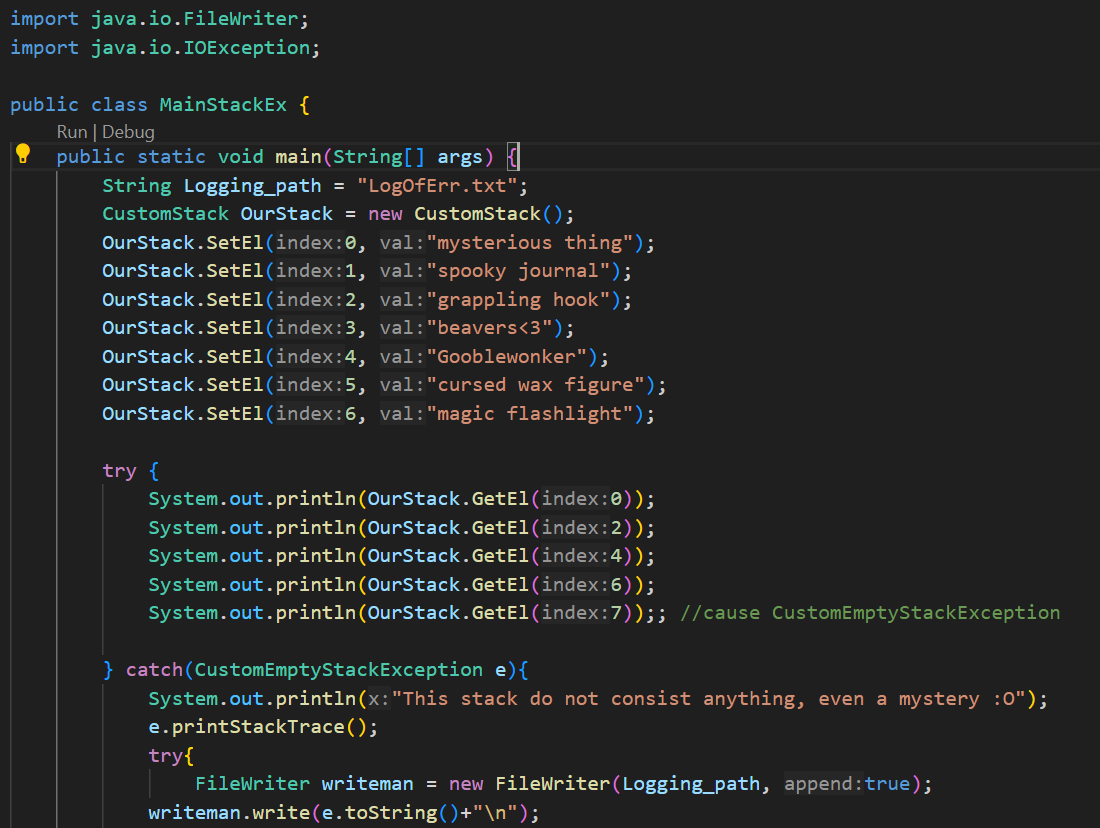
CustomEmptyStackException.

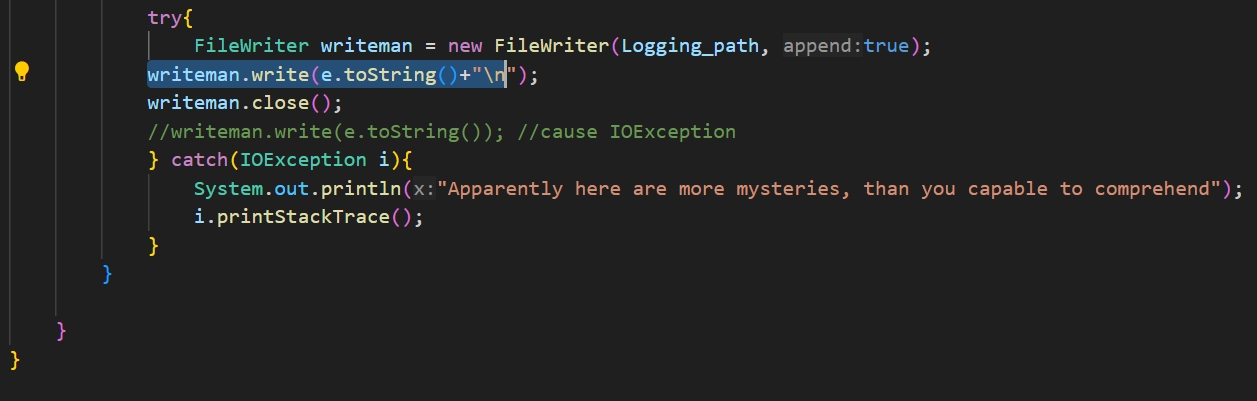
****

Создадим свой собственный класс исключений, который наследует класс Exception (все исключения, которые мы создаём, должны наследовать его) и создадим для него лишь конструктор с параметром, которым будет являться сообщение об ошибке. Конструктор нашего класса-исключения вызывает родительский конструктор класса Exception.

****

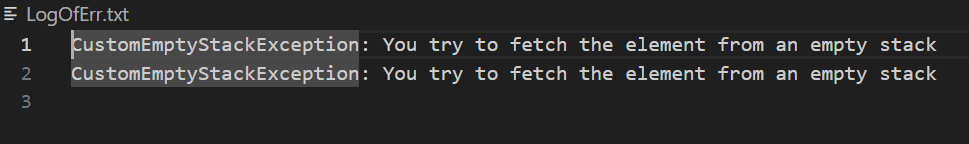
Теперь создадим наш класс, для создания стэков, где будем хранить строки и назначать количество элементов в массиве (стэке) при инициализации. Стоит обратить внимание на метод, который может вызывать исключение. В нём мы вначале прописываем с помощью ключевого слова throws, какое исключение он выкидывает, а затем условие при котором это происходит. В данном случае — когда мы обращаемся к пустому элементу массива (само исключение себя не выкинет и нам нужно указать, когда это должно случаться).

****

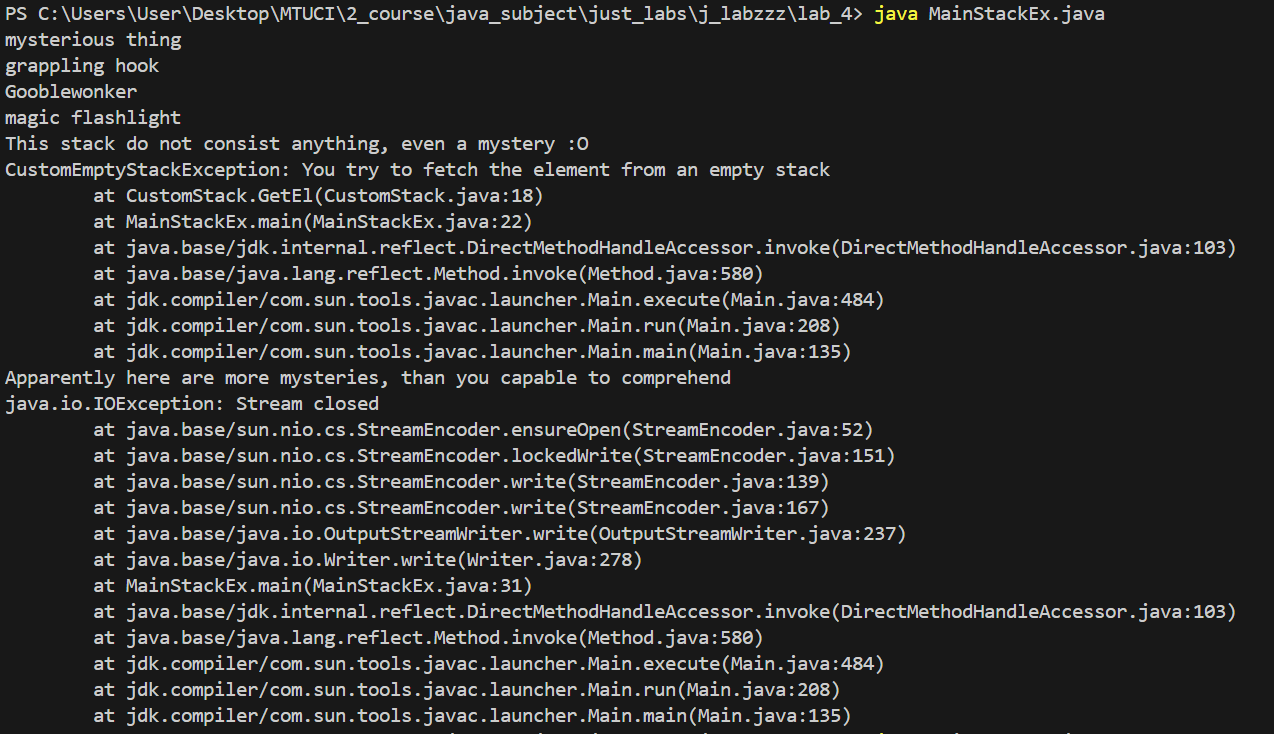
****

И наконец напишем файл мэйн, в котором заполним объект нашего стока значениями, а затем попытаемся их вывести и при возникновении исключений обработаем известным нам способом.

Помимо прочего мы записываем исключение, касающееся попытки извлечения из стэка пустого значения с помощью известного нам из второго задания этой лабораторной класса FileWriter, в отдельный текстовый файл. Таким образом мы реализуем логгирование.

****

Для наглядности продемонстрируем, как выглядят мои исключения в консоли, когда они возникают при выполнении программы.

****

**Вывод:** Мы ознакомились со структурой и иерархией исключений в джава и с их практическим применением.

А также поняли как они работают и научились взаимодействовать с ними, обрабатывая.

Более того, мы научились создавать собственные исключения и контролировать их поведение.