Министерство цифрового развития, связи и массовых коммуникаций

Российской Федерации

Ордена Трудового Красного Знамени

Федерально государственное бюджетное образовательное учреждение

высшего образования

МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ СВЯЗИ И ИНФОРМАТИКИ

Кафедра «Математической кибернетики и информационных технологий»

Лабораторная работа №4

По дисциплине

«Информационные технологии и программирование»

Выполнил:

Студент группы БВТ2203

Бородин К.Н.

Москва 2023

ЗАДАНИЯ

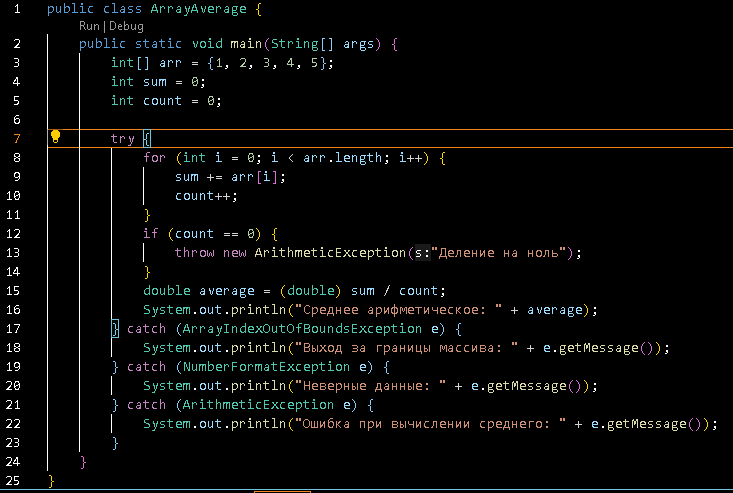
1. Необходимо написать программу, которая будет находить среднее арифметическое элементов массива. При этом программа должна обрабатывать ошибки, связанные с выходом за границы массива и неверными данными (например, если элемент массива не является числом).

2. Необходимо написать программу, которая будет копировать содержимое одного файла в другой. При этом программа должна обрабатывать возможные ошибки, связанные с чтением и записью файлов.

3. Создайте Java-проект для работы с исключениями. Для задачи напишите свой собственный класс для обработки исключений. Создайте обработчик исключений, который логирует информацию о каждом выброшенном исключении в текстовый файл. Реализовать следующую задачу. Создать класс CustomNumberFormatException, который будет использоваться для обработки исключения NumberFormatException. Реализовать программу, которая пытается преобразовать строкув число (Integer.parseInt()), и, если строка не является числом, выбрасывайте исключение CustomNumberformatException.

ХОД РАБОТЫ.

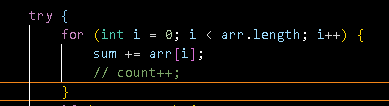
Создаю массив, иницилизирую переменный для подсчёта суммы и количества элементов. Далее пробую посчитать среднее арифметечиское. Если всего чисел ноль, то выбрасываю исключение. Далее обрабатываю исключения, связанные с выходом за пределы массива, наличием не числовых данных или другой арифметической ошибкой, рисунок 1.

  
Рисунок 1 – нахождение среднего арифметического.

Данная программа даёт корректный результат, рисунок 2.

  
Рисунок 2- правильный результат.

Убираю подсчёт количества элементов, чтобы продемонстрировать исключение, рисунок 3.

  
Рисунок 3 – изменение программы.

Результат программу представлен на рисунке 4.

  
Рисунок 4- результат программы.

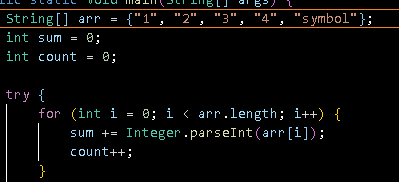
Меняю программу для получения исключения, рисунок 5.

  
Рисунок 5 – изменение программы.

Результат работы программы представлен на рисунке 6.

  
Рисунок 6 – результат работы программы.

Снова меняю программу для демонстрации исключения, рисунок 7.

  
Рисунок 7 – изменение программы.

Результат работы программы представлен на рисунке 8

  
Рисунок 8 – результат работы программы.

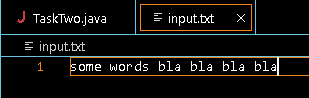
Создаю программу для записи данных из одного файла в другой. В начале инициализирую пути для файлов. Затем в обработчике исключения указываю файлы, которые будут закрыты после его работы. Внутри пробую перезаписать файл. Если не получается, ловлю исключения, связанные с чтением/записью, рисунок 9.

  
Рисунок 9 – программа для перезаписи данных.

Поскольку файла не было, то было поймано исключение. Рисунок 10.

  
Рисунок 10 – исключение.

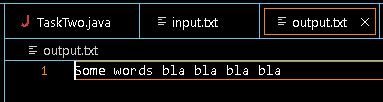
Создаю первый файл, рисунок 11.

  
Рисунок 11 – создание файла.

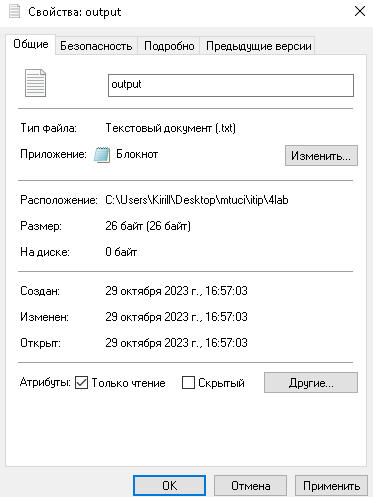
Программа завершилась успешно, рисунок 12.

  
Рисунок 12 – успешное завершение программы.

Файл действительно записан, рисунок 13.

  
Рисунок 13 – перезаписанный файл.

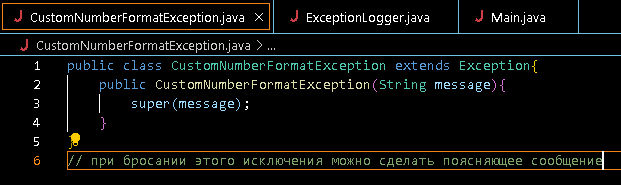
Теперь я разрешаю перезаписанный файл открывать только для чтения, рисунок 14.

  
Рисунок 14 – введение ограничений на запись файла.

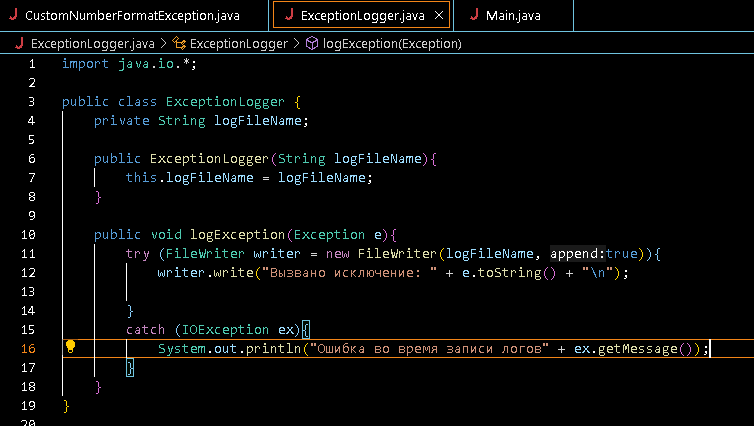
Вызвано исключение, рисунок 15.

  
Рисунок 15 – вызов исключения.

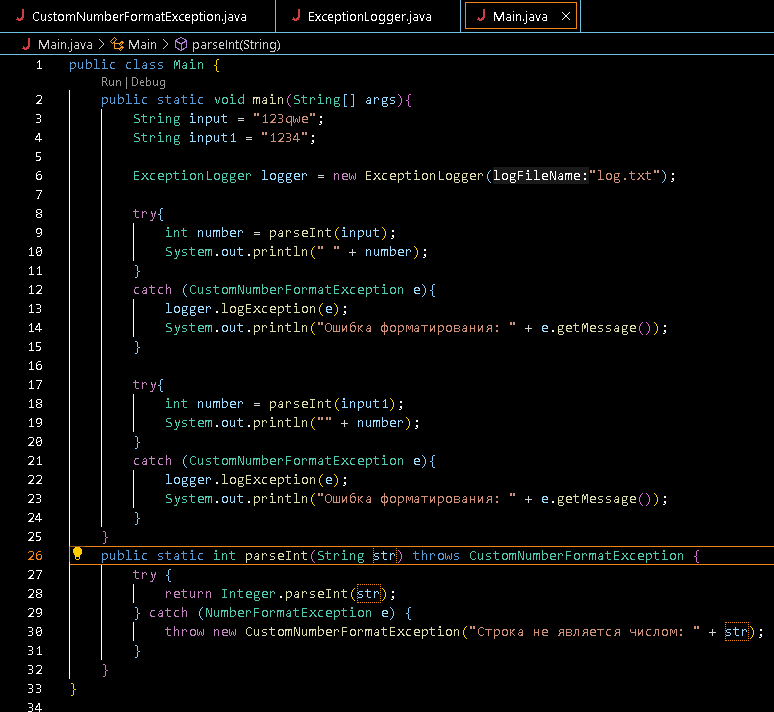
Создаю класс для обработки собственного исключения, рисунок 16.

  
Рисунок 16 – созданию собственного исключения.

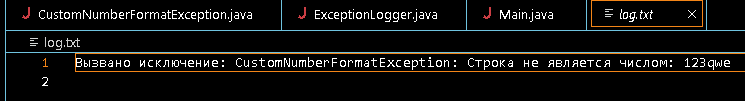
Создаю обработчик исключения, логирующий информацию, рисунок 17.

  
Рисунок 17 – создание обработчика исключений.

Выполняю задание. Создаю строку с ошибкой и без. Далее создаю экзмемпляр обработчика исключений. В коде есть функция для преобразования строки в число, при этом она бросает наше кастомное исключение, если есть ошибки. Далее пробую с помощью этой функции конвертировать строку, и если появляется исключения, то оно логируется. Программа представлена на рисунке 18.

  
Рисунок 18 – программа.

Обработчик исключения залогировал информацию, рисунок 19.

  
Рисунок 19 – логированная информация.