Задание 1

Мной был создан простой программный модуль, в котором возникает ошибка деления на ноль.

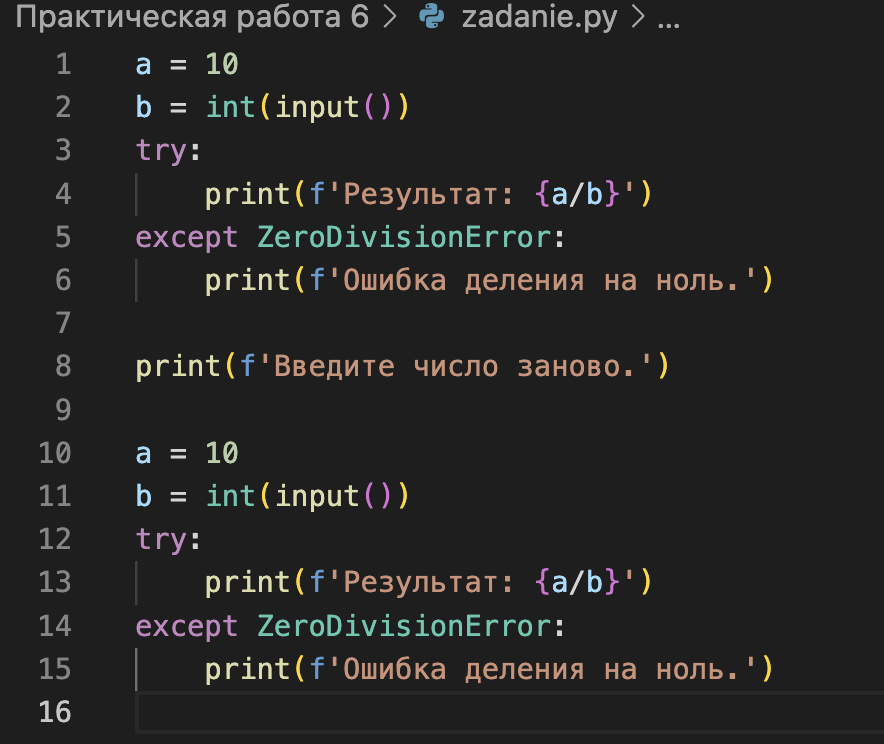


Рисунок 1 – код

Была реализована обработка исключений с помощью конструкции try-except, чтобы предотвратить аварийное завершение программы. При возникновении ошибке программный модуль просит ввести число заново.

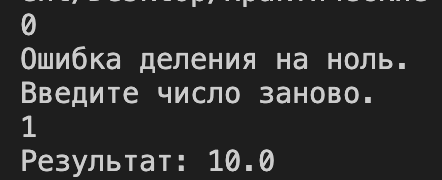


Рисунок 2 – результат

Задание 2

Мной создано несколько примеров обработки различных исключений, такие как ZeroDivisionError, ValueError, IndexError.

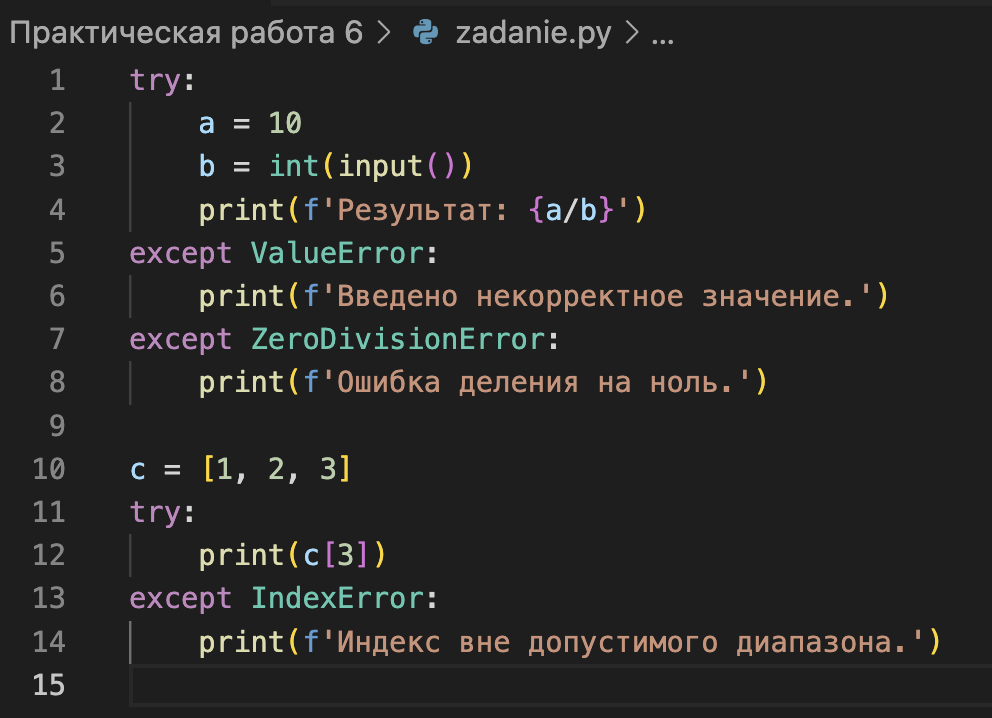


Рисунок 3 – примеры обработки исключений в коде

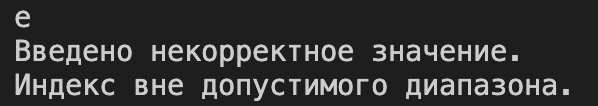


Рисунок 4 – результат обработки исключений

Задание 3

Мной был создан пользовательский класс исключений для обработки специфических ситуаций в программном проекте и написан пример кода, который генерирует и обрабатывает это исключение.

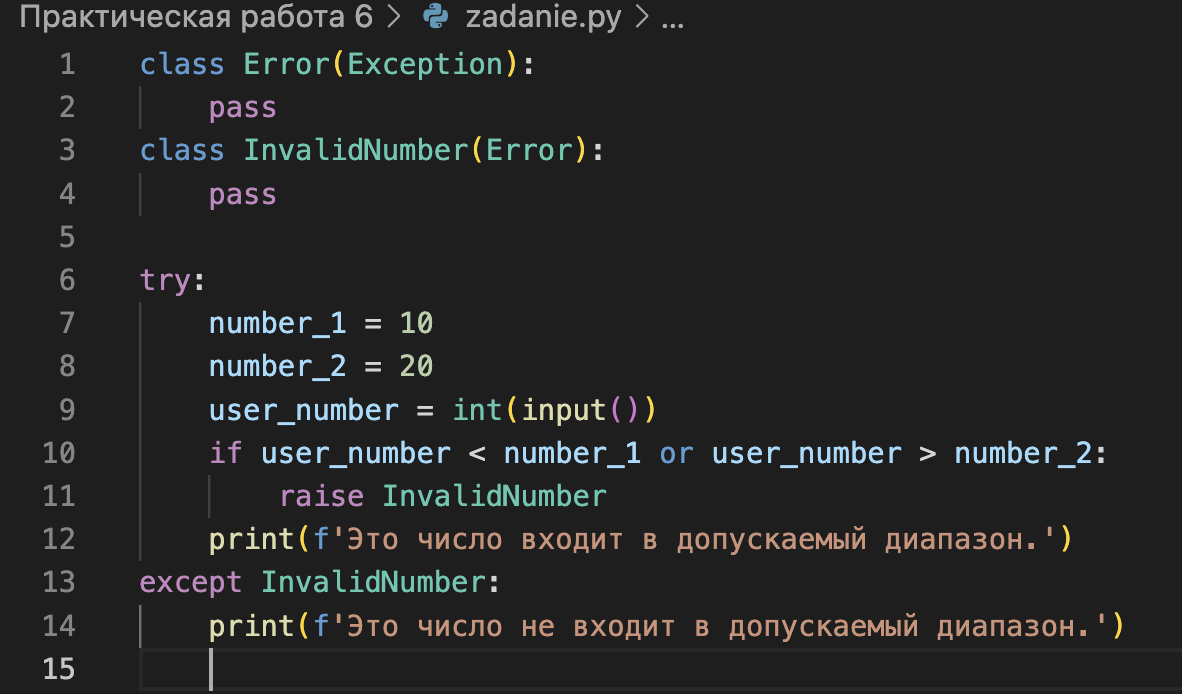


Рисунок 5 – пользовательские классы

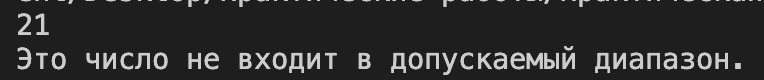


Рисунок 6 – обработка исключения

Почему пользовательские классы так важны и необходимы в программировании? Пользовательские классы— это определенные пользователем чертежи, из которых создаются объекты. Эти чертежи определяют структуру и поведение объектов, созданных из них. Они инкапсулируют данные для объекта и методы для манипулирования этими данными, следуя принципам инкапсуляции и абстракции.

Задание 4

Мной реализован механизм логирования исключений в файл py\_log.log для их дальнейшего анализа. Была проверена корректность логирования и каждая ошибка записывается с указанием времени, типа ошибки и контекста.

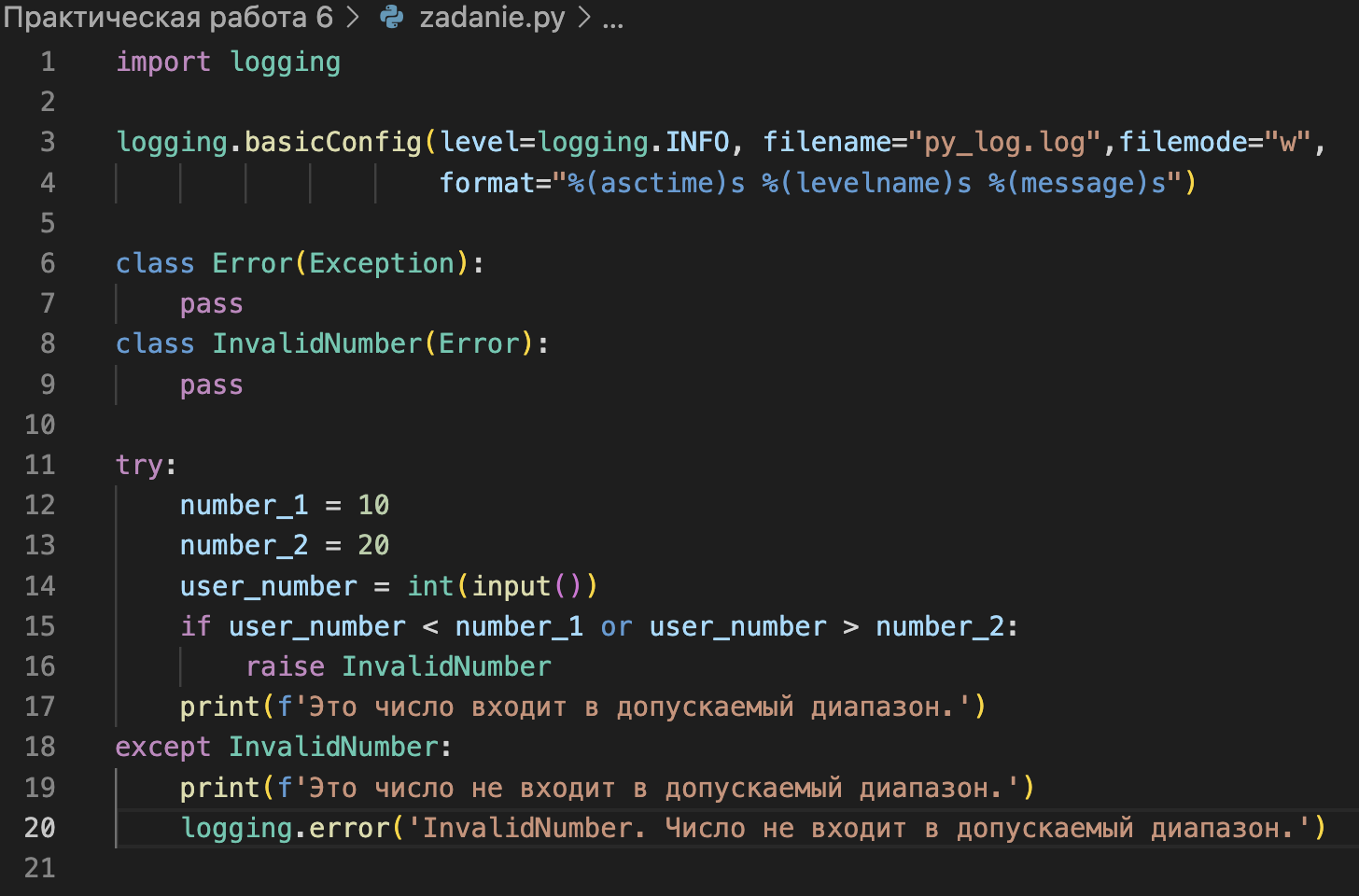


Рисунок 7 – добавлен механизм логирования

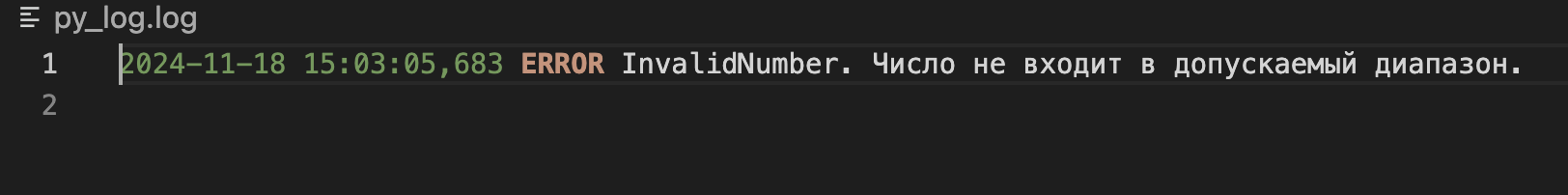


Рисунок 8 – проверка записи логов

Задание 5

Мной был написан код, который использует многопоточность и реализует обработку исключений в многопоточной среде. Исключения в одном потоке не приводят к завершению программы и других потоков.

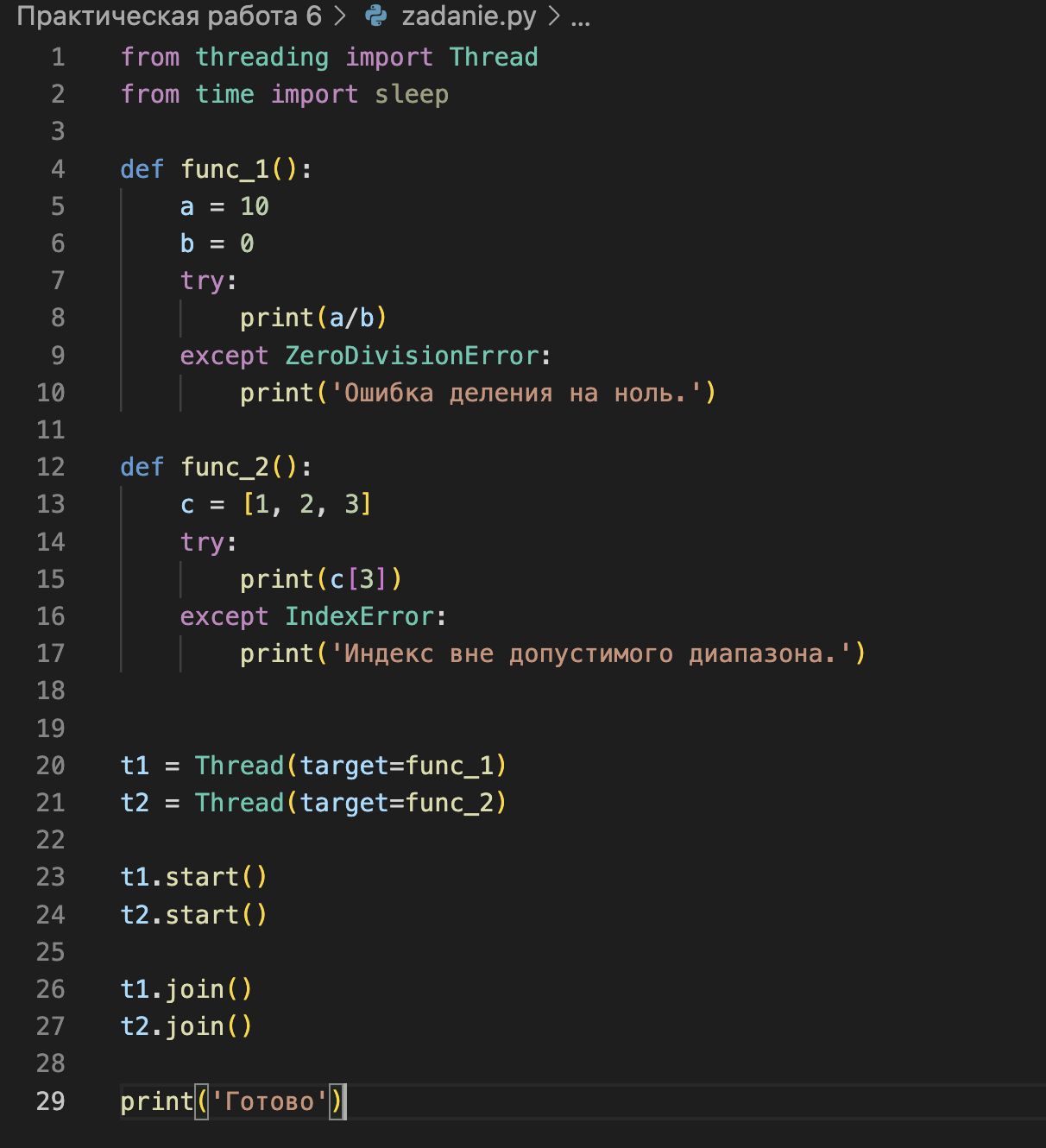


Рисунок 9 – многопоточность