

Факультет/Институт	Информационных технологий (наименование факультета/ Института)
Направление/специальность подготовки:	Информационные системы и технологии (код и наименование направления /специальности подготовки)
Форма обучения:	очная (очная, очно-заочная, заочная)

по дисциплине Тестирование информационных систем
(наименование дисциплины)

Группа ДКИП-311

Преподаватель Сибирев Иван Валерьевич _____
(ФИО) (подпись)

Москва 2025 г.

Практическая работа №4

Тема: Анализ и обеспечение обработки исключительных ситуаций.

Цель работы: Изучение способов обработки исключительных ситуаций.

Используемые типы исключений в программе:

1. FileNotFoundException - файл не найден
2. IOException - ошибка ввода и вывода
3. UnauthorizedAccessException - нет прав доступа к файлу
4. FormatException - неверный формат входных данных
5. DivideByZeroException - деление на ноль
6. ArgumentException - неверный
7. Exception - общее исключение для перехвата непредвиденных ошибок

Работа программы

При попытке открыть несуществующий файл программа перехватывает исключение FileNotFoundException и выводит сообщение:

```
Введите имя входного файла: 1.txt
Ошибка: Файл 1.txt не найден
```

При делении на ноль программа перехватывает исключение DivideByZeroException:

```
Введите первое число:
23
Введите операцию (+, -, *, /):
/
Введите второе число:
0
Ошибка: Нельзя делить на ноль
```

При вводе некорректных данных программа обрабатывает исключение FormatException:

```
Введите первое число:
aaa
Ошибка формата: Неверный формат первого числа
```

```
Введите первое число:  
23  
Введите операцию (+, -, *, /):  
+  
Введите второе число:  
gg  
Ошибка формата: Неверный формат второго числа
```

5. Заключение

В ходе выполнения практической работы были изучены методы обработки исключительных ситуаций в языке программирования C#. Разработана программа, которая выполняет различные типы исключений

Ответы на контрольные вопросы

1. Что такое исключительная ситуация?

Исключительная ситуация (исключение) - это аномалия, которая может возникать во время выполнения программы и которую трудно предусмотреть во время программирования. К числу таких возможных исключений относятся попытки подключения к базе данных, которой больше не существует, попытки открытия поврежденного файла или попытки установки связи с машиной, которая в текущий момент находится в автономном режиме.

2. Как можно обработать исключительную ситуацию?

Программирование со структурированной обработкой исключений подразумевает использование четырех следующих связанных между собой сущностей:

- тип класса, который представляет детали исключения;
- член, способный генерировать (throw) в вызывающем коде экземпляр класса исключения при соответствующих обстоятельствах;
- блок кода на вызывающей стороне, ответственный за обращение к члену, в котором может произойти исключение;
- блок кода на вызывающей стороне, который будет обрабатывать (или перехватывать (catch)) исключение в случае его возникновения

Для того чтобы справиться с возможными ошибочными ситуациями в коде C#, программа обычно делится на блоки трех разных типов:

- Блоки try инкапсулируют код, формирующий часть нормальных действий программы, которые потенциально могут столкнуться с серьезными ошибочными ситуациями.

- Блоки `catch` инкапсулируют код, который обрабатывает ошибочные ситуации, происходящие в коде блока `try`. Это также удобное место для протоколирования ошибок.
- Блоки `finally` инкапсулируют код, очищающий любые ресурсы или выполняющий другие действия, которые обычно нужно выполнить в конце блоков `try` или `catch`. Важно понимать, что этот блок выполняется независимо от того, сгенерировано исключение или нет