

## **Практическое занятие № 9**

**Тема практического занятия:** Исключительные ситуации и их обработка.

**Цель практического занятия:** приобретение знаний и практических навыков для правильной обработки исключений в своих программах, повышая их надежность, устойчивость и читаемость.

**В результате выполнения данной работы обучающийся должен уметь:**

1. Обрабатывать исключения в коде программы с целью оповещения пользователя о возникновении ошибки или некорректного состояния.
2. Правильно обрабатывать исключения в коде.
3. Проектировать и использовать свои собственные исключения

**знать:**

1. Что такое исключительные ситуации и почему они возникают.
2. Синтаксис и структуру обработки исключений.
3. Важность обработки исключений.

**Перечень оборудования, необходимого для выполнения задания:**

- Автоматизированные рабочие места по количеству обучающихся (процессор Intel Core i7 или аналогичный, БП 700 Вт, 32 Гб ОЗУ, SSD 512 Гб, HDD 4 ТБ SATA 7200 rpm, RTX 3060 12GB);
  - Автоматизированное рабочее место преподавателя (процессор Intel Core i7 или аналогичный, БП 700 Вт, 32 Гб ОЗУ, SSD 512 Гб, HDD 4 ТБ SATA 7200 rpm, RTX 3060 12GB);
  - Монитор 34", изогнутый, 3440x1440, 6 ms, 178°/178°, 300 cd/m2, 20M:1, HDMI, DP, USB, регулировка по высоте (1 монитор на одно рабочее место).
  - Интерактивная панель 86" с OPS ПК
- Программное обеспечение:
- Microsoft Visual Studio.

**Общие теоретические сведения:**

Исключительная ситуация (или исключение) - это непредвиденное событие или ошибка, которая может возникнуть во время выполнения программы и привести к ее некорректному поведению или остановке.

Исключительные ситуации могут быть вызваны различными причинами, такими как ошибка ввода-вывода, деление на ноль, отсутствие ресурсов, выход за границы массива и другие некорректные операции. Когда исключительная ситуация возникает, программа может выйти из нормального потока выполнения и перейти к специальному блоку кода, предназначенному для ее обработки.

Цель обработки исключительных ситуаций состоит в том, чтобы предотвратить аварийное завершение программы и предоставить возможность корректно реагировать на ошибочные ситуации. Обработка исключительных ситуаций позволяет организовать контроль над ошибками и принять меры для

восстановления нормального состояния программы или надлежащего уведомления пользователей об ошибке.

Одним из способов работы с исключительными ситуациями является использование механизма исключений, предоставляемого языками программирования.

Использование исключительных ситуаций позволяет писать более надежный и устойчивый код, предупреждая некорректное поведение программы, и позволяет программисту контролировать и обрабатывать ошибки, упрощая отладку и улучшая общий опыт пользователей

### **Задание:**

1. Модифицировать библиотеку классов, разработанное во время практического занятия №8, добавив необходимые исключительные ситуации, не являющимися системными (отрицательное число достопримечательностей, нецелое число человек и т.п.).
2. Доработайте приложение, разработанное во время практического занятия №8, добавив туда обработку исключительных ситуаций: как системных, так и собственных.

### **Указания по технике безопасности:**

Инструкция по технике безопасности при работе в лаборатории, оборудованной компьютерной техникой.

### **Технология выполнения работы (этапы, последовательность действий):**

1. Проанализируйте поля и методы класса на предмет возможных возникновений ситуаций, вызванных некорректными действиями пользователей.
  2. Добавьте собственные исключения в код библиотеки.
  3. Пересоберите библиотеку, убедитесь в успешности сборки.
  4. Проанализируйте приложение на предмет возможности возникновения ваших и системных исключительных ситуаций.
  5. Добавьте обработку исключительных ситуаций там, где это необходимо.
- Пример кода с обработкой системной исключительной ситуации (деление на 0):

```

static void Main(string[] args)
{
    Random rnd = new Random();
    int x = rnd.Next(-1000, 1001);
    Console.Write("Введите число: ");
    try
    {
        int y = int.Parse(Console.ReadLine());
        Console.WriteLine($"Остаток от деления равен {x % y}");
    }
    catch
    {
        Console.WriteLine("Произошла ошибка!");
    }
    Console.ReadKey();
}

```

Пример кода с обработкой исключительной ситуации, генерируемой разработчиком (использование отрицательного числа там, где это невозможно):

```

static void Main(string[] args)
{
    try
    {
        Console.Write("Введите ширину участка: ");
        int width = int.Parse(Console.ReadLine());
        Console.Write("Введите длину участка: ");
        int length = int.Parse(Console.ReadLine());
        if (width <= 0 || length <= 0)
            throw new Exception("Размер не может быть меньше нуля!");
        Console.WriteLine($"Площадь участка - {width*length}");
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
    }
    Console.ReadKey();
}

```

6. При проектировании обработки исключительных ситуаций учтите, что пользователю необходимо получить сведения о том, в чем именно заключается ошибка: какое именно поле задано некорректно, что требуется изменить. Однако если вы будете выдавать много текстовых сообщений, то интерфейс не будет эргономичным. Поэтому продумайте, как именно вы будете уведомлять пользователя.

#### Требование к отчету:

1. В библиотеку классов добавлены собственные исключения.

2. Добавленные собственные исключения логичны, целесообразны и покрывают все возможные некорректные ситуации.
3. В приложении осуществлена обработка всех исключительных ситуаций, которые могут быть вызваны действиями пользователя.

### **Контрольные вопросы:**

1. Какие типы исключительных ситуаций могут возникнуть в программном коде?
2. Какие возможности языка программирования предоставляются для обработки исключительных ситуаций?
3. Какой синтаксис использовать для создания исключительных ситуаций в коде?
4. Чем отличается использование исключительных ситуаций от использования условных операторов?
5. Что такое блок try-catch и как он используется для обработки исключений?
6. Какое действие выполняется при возникновении исключительной ситуации и какие операторы выполняются в блоке catch?
7. Как обработать исключительную ситуацию, если она не может быть решена локально в текущем блоке?
8. Что такое блок finally и для чего он используется?
9. Какие приемы можно применить для предотвращения возникновения исключительных ситуаций?

### **Основные и дополнительные источники, электронные ресурсы:**

1. Подбельский, В. В. Язык C#. Базовый курс: учебное пособие / В. В. Подбельский. - 2-е изд., перераб. и доп. - Москва: Финансы и статистика, 2022. - 408 с. - ISBN 978-5-00184-079-4. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1913989>
2. Гуриков, С. Р. Основы алгоритмизации и программирования на Python: учебное пособие / С.Р. Гуриков. — Москва: ИНФРА-М, 2023. — 343 с. — (Среднее профессиональное образование). - ISBN 978-5-16-016906-4. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1927269>.