

Лабораторная работа №4

Изучение процесса разработки и конфигурирования классов исключений

1 Цель работы

1.1 Изучить процесс создания и применения собственных классов исключений и настройки стандартных исключений на C#.

2 Литература

2.1 <https://metanit.com/sharp/tutorial> – гл.4.

3 Подготовка к работе

3.1 Повторить теоретический материал (см.п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Генерация исключения с требуемым текстом исключения

5.1.1 Создать класс для хранения идентификатора, логина и пароля пользователя.

Для хранения данных в классе использовать закрытые поля.

Для каждого поля создать свойство на чтение и запись, используя инкапсуляцию полей (Быстрые действия и рефакторинг – Инкапсулировать поле).

Идентификатор должен изменяться, если значение value > 0.

Пароль должен изменяться, если длина значения value не больше 20 символов.

Логин должен изменяться, если длина значения value не меньше 1 символа и не больше 30 символов.

5.1.2 Реализовать генерацию исключения с текстом «Идентификатор должен быть положительным» при попытке изменить на некорректное значение поля идентификатор.

Для генерации исключения использовать следующий код:

```
throw new Исключение(текст исключения);
```

5.1.3 Проверить вызов исключения, создав объект класса Пользователь и присвоив ему корректные и некорректные значения идентификатора. Перехват исключений требуется выполнять в вызывающем методе. При перехвате выводить сообщение исключения.

5.2 Создание дочернего класса исключения

5.2.1 Создать класс StringArgumentException, наследника класса ArgumentException, для описания исключения несоответствия длины строки. В классе при вызове конструктора по умолчанию должен вызываться конструктор базового класса с текстом «Длина строки не соответствует ограничениям».

5.2.2 Реализовать генерацию исключений StringArgumentException при указании пароля / логина неверной длины в соответствующих свойствах класса Пользователь.

5.2.3 Проверить вызов исключения, создав объект класса Пользователь и присвоив ему корректные и некорректные по длине значения логина и пароля. Перехват исключений требуется выполнять в вызывающем методе. При перехвате выводить сообщение исключения.

5.3 Переопределение стандартных свойств класса исключения

5.3.1 Написать в классе `StringArgumentException` `override` и выбрать для переопределения свойство `HelpLink`. Изменить его так, чтобы свойство возвращало ссылку или email, по которому можно получить консультацию о возникшей ошибке.

5.3.2 Вывести на экран значение `HelpLink` при перехвате исключения.

5.4 Создание у класса исключения конструктора с параметрами

5.4.1 Добавить в класс `StringArgumentException` конструктор с двумя параметрами `int minLength` и `int maxLength` и изменить в нем свойство `Message` так, чтобы в сообщении кроме текста «Длина строки не соответствует ограничениям» выводилось требование к длине строки «Длина строки должна быть от min до max» (вместо min и max должны браться переданные в параметрах конструктора значения).

5.4.2 Реализовать генерацию исключений `StringArgumentException` при указании пароля / логина неверной длины в соответствующих свойствах класса Пользователь. При создании исключений использовать созданный в п.5.4.1 конструктор.

5.4.3 Проверить вызов исключения, создав объект класса Пользователь и присвоив ему корректные и некорректные значения логина и пароля. Перехват исключений требуется выполнять в вызывающем методе. При перехвате выводить сообщение исключения.

5.5 Добавление свойств в класс исключения

5.5.1 Добавить в класс `StringArgumentException`:

- автореализуемое свойство только на чтение для возврата кода ошибки
- конструктор с параметрами `string value`, `int minLength` и `int maxLength`, в котором изменять значение кода ошибки (код ошибки 1: длина переданной строки меньше минимальной, код ошибки 2: длина переданной строки больше максимальной).

5.5.2 Реализовать генерацию исключений `StringArgumentException` при указании пароля / логина неверной длины в соответствующих свойствах класса Пользователь. При создании исключений использовать созданный в п.5.5.1 конструктор.

5.5.3 Проверить вызов исключения, создав объект класса Пользователь и присвоив ему корректные и некорректные значения логина и пароля. Перехват исключений требуется выполнять в вызывающем методе. При перехвате выводить код ошибки и сообщение исключения.

6 Порядок выполнения работы

6.1 Выполнить все задания из п.3 в одном решении LabWork4. Возможные ошибки требуется обрабатывать. Выполнить форматирование и рефакторинг кода.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Какой класс в C# является базовым для всех классов исключений?

8.2 Что возвращает свойство `Message` у классов исключений?

8.3 Какое свойство у классов исключений возвращает стек вызова, приведшего к появлению исключения?

8.4 Для чего используется свойство `HelpLink` у классов исключений?

8.5 В чем отличие между типами исключений `SystemException` и `ApplicationException`?