**Структура обработки исключений в .NET, проектирование собственных типов исключений**

Задание 1 Проверка строки на пустое значение. Класс StringProcessor содержит метод ValidateInput(string input). Если строка пустая или null, выбрасывается EmptyStringException.

Листинг программы:

public class EmptyStringException : Exception

{

public EmptyStringException() : base("Строка не должна быть пустой или null.") { }

public EmptyStringException(string message) : base(message) { }

public EmptyStringException(string message, Exception innerException)

: base(message, innerException) { }

}

public class StringProcessor

{

public void ValidateInput(string input)

{

if (string.IsNullOrEmpty(input))

{

throw new EmptyStringException();

}

}

}

class Program

{

static void Main(string[] args)

{

StringProcessor processor = new StringProcessor();

string input = "";

try

{

processor.ValidateInput(input);

}

catch (EmptyStringException ex)

{

Console.WriteLine($"Ошибка: {ex.Message}");

}

input = "Hello, World!";

try

{

processor.ValidateInput(input);

Console.WriteLine("Строка валидна.");

}

catch (EmptyStringException ex)

{

Console.WriteLine($"Ошибка: {ex.Message}");

}

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| input: "", input: "Hello, World!" | Ошибка: Строка не должна быть пустой или null, Строка валидна. |

Анализ результатов:



Рисунок 1.1 – Результат работы программы