Лабораторная работа 4. Простейшие

классы

Теоретический материал: глава 4, раздел «Обработка исключительных ситуаций», глава

5. Каждый разрабатываемый класс должен, как правило, содержать следующие

элементы: скрытые поля, конструкторы с параметрами и без параметров, методы,

свойства. Методы и свойства должны обеспечивать непротиворечивый, полный,

минимальный и удобный интерфейс класса. При возникновении ошибок должны

выбрасываться исключения. В программе должна выполняться проверка всех

разработанных элементов класса.

Вариант 4

Построить описание класса, содержащего информацию о почтовом адресе организации. Предусмотреть возможность раздельного изменения составных частей адреса и проверки допустимости вводимых значений. В случае недопустимых значений полей выбрасываются исключения. Написать программу, демонстрирующую все разработанные элементы класса.

Задание:

using System;

namespace OrganizationAddress

{

// Класс исключения для недопустимых значений адреса

public class InvalidAddressException : Exception

{

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

}

// Класс почтового адреса

public class PostalAddress

{

private string street;

private string city;

private string state;

private string postalCode;

public string Street

{

get => street;

set

{

if (string.IsNullOrWhiteSpace(value))

throw new InvalidAddressException("Улица не может быть пустой.");

street = value;

}

}

public string City

{

get => city;

set

{

if (string.IsNullOrWhiteSpace(value))

throw new InvalidAddressException("Город не может быть пустым.");

city = value;

}

}

public string State

{

get => state;

set

{

if (string.IsNullOrWhiteSpace(value))

throw new InvalidAddressException("Область не может быть пустой.");

state = value;

}

}

public string PostalCode

{

get => postalCode;

set

{

if (string.IsNullOrWhiteSpace(value))

throw new InvalidAddressException("Почтовый индекс не может быть пустым.");

if (value.Length != 5 || !IsAllDigits(value))

throw new InvalidAddressException("Почтовый индекс должен состоять из 5 цифр.");

postalCode = value;

}

}

public PostalAddress(string street, string city, string state, string postalCode)

{

Street = street;

City = city;

State = state;

PostalCode = postalCode;

}

private bool IsAllDigits(string s)

{

foreach (char c in s)

{

if (!char.IsDigit(c))

return false;

}

return true;

}

public override string ToString()

{

return $"{Street}, {City}, {State}, {PostalCode}";

}

}

class Program

{

static void Main()

{

try

{

PostalAddress address = new PostalAddress("Ленина", "Москва", "Москва", "10100");

Console.WriteLine("Исходный адрес: " + address);

// Изменение адреса с проверкой

address.Street = "Пушкина";

address.City = "Санкт-Петербург";

address.State = "Санкт-Петербург";

address.PostalCode = "19000";

Console.WriteLine("Обновленный адрес: " + address);

// Пример недопустимого значения

address.PostalCode = "1234"; // Это вызовет исключение

}

catch (InvalidAddressException ex)

{

Console.WriteLine("Ошибка: " + ex.Message);

}

}

}

}

