**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ** **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** **«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ** **“СИНЕРГИЯ”»**

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт** |  | Синергия |
|  |  | (наименование факультета/ Института) |
| **Направление/специальность** |  | Информационные системы и программирование |
| **подготовки:** |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  |  |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

**Отчет по лабораторной работе №4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **на тему** |  | Массивы и строки | | |
|  |  | (наименование темы) | | |
|  |  |  | | |
| **по дисциплине** | | |  | Разработка программных модулей |
|  | | |  | (наименование дисциплины) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | Даякаева Исманат |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | ДКИП–202прог |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  | Сиберев И.В. |  |  |
|  |  | (ФИО) |  | (подпись) |

**Москва 2025 г.**

**Лабораторная работа №4**

**Тема: Массивы и строки**

**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.

Код:

**using System;**

**public class Address**

**{**

**public string City { get; set; }**

**public string Street { get; set; }**

**public string HouseNumber { get; set; }**

**public Address(string city, string street, string houseNumber)**

**{**

**if (string.IsNullOrWhiteSpace(city))**

**{**

**throw new ArgumentException("City cannot be empty.");**

**}**

**if (string.IsNullOrWhiteSpace(street))**

**{**

**throw new ArgumentException("Street cannot be empty.");**

**}**

**if (string.IsNullOrWhiteSpace(houseNumber))**

**{**

**throw new ArgumentException("House number cannot be empty.");**

**}**

**City = city;**

**Street = street;**

**HouseNumber = houseNumber;**

**}**

**public override string ToString()**

**{**

**return $"{City}, {Street}, {HouseNumber}";**

**}**

**}**

**public class Program**

**{**

**public static void Main(string[] args)**

**{**

**try**

**{**

**Address myAddress = new Address("Москва", "Тверская", "15");**

**Console.WriteLine(myAddress);**

**myAddress.City = "Санкт-Петербург";**

**Console.WriteLine(myAddress);**

**// Test for invalid input.  Note: This code will throw an exception and halt execution if placed inside the main Try block.  Moved to another try-catch**

**}**

**catch (ArgumentException e)**

**{**

**Console.WriteLine($"Error: {e.Message}");**

**}**

**try**

**{**

**Address invalidAddress = new Address("", "Some Street", "123"); //City is empty**

**}**

**catch (ArgumentException e)**

**{**

**Console.WriteLine($"Error: {e.Message}");**

**}**

**Console.WriteLine("Press any key to exit."); //Informative message**

**Console.ReadKey();**

**}**

**}**





