**Лабораторная работа по курсу**

**«Платформа Microsoft .NET и язык программирования C#» №9**

**Тема: Индексаторы и свойства.**

**Цель:**

Закрепить у слушателей практические навыки и теоретические знания для работы классами и объектами, индексаторами и свойствами. Научиться перегружать индексаторы.

**Необходимые инструменты:** MS Visual Studio 2010

**Документация:** Конспект, Литература

**Ориентировочное время исполнения:** 2 часа.

**Задание 1.**

В С # индексация начинается с нуля, но в некоторых языках программирования это не так. Например, в Turbo Pascal индексация массиве начинается с 1. Напишите класс **SmartArray**, который позволяет работать с массивом такого типа, в котором индексный диапазон устанавливается пользователем. Например, в диапазоне от 6 до 10, или от -9 до 15.

**Подсказка:**  Установка диапазонов в классе обеспечивается с использованием соответствующих свойств.

**Задание 2.**

Создать класс **Map**, позволяющий работать с **ДИНАМИЧЕСКИМ** словарем. **Словарь (dictionary)**  представляет собой сложную структуру данных, позволяющую обеспечить доступ к элементам по ключу. Требования к классу представлены в приложении 1.

Протестировать созданный класс в методе Main().

**ПРИЛОЖЕНИЕ 1**

**public class Map**

**{**

**// Сводка:**

**// Инициализирует новый пустой экземпляр класса**

**public Map ();**

**// Сводка:**

**// Возвращает число пар "ключ-значение", содержащихся в словаре**

**public int Count { get; }**

**// Сводка:**

**// Получает массив, содержащий ключи в словаре**

**public string[] Keys { get; }**

**// Сводка:**

**// Получает массив, содержащий значения в словаре**

**public double [] Values { get; }**

**// Сводка:**

**// Возвращает или задает значение, связанное с указанным ключом.**

**// Параметры:**

**// key:**

**// Ключ, значение которого требуется получить или задать.**

**// Возвращает:**

**// Значение, связанное с указанным ключом.**

**// Если указанный ключ не найден, операция**

**// генерирует исключение KeyNotExistException,**

**public double this[string key] { get; set; }**

**// Сводка:**

**// Добавляет указанные ключ и значение в словарь.**

**// Параметры:**

**// key:**

**// Ключ добавляемого элемента.**

**// value:**

**// Добавляемое значение элемента.**

**// Исключения:**

**// System.ArgumentException:**

**// Элемент с таким ключом уже существует в словаре**

**public void Add(string key, double value);**

**// Сводка:**

**// Удаляет все ключи и значения из словаря**

**public void Clear();**

**// Сводка:**

**// Определяет, содержится ли указанный ключ в словаре**

**// Параметры:**

**// key:**

**// Ключ, который требуется найти в**

**// Возвращает:**

**// true, если словарь содержит элемент с указанным ключом,**

**// в противном случае — false.**

**public bool ContainsKey(string key);**

**// Сводка:**

**// Определяет, содержит ли словарь указанное значение.**

**// Параметры:**

**// value:**

**// Значение, которое требуется найти в словаре**

**// Возвращает:**

**// Значение true, если элемент с указанным значением,**

**// в противном случае — значение false.**

**public bool ContainsValue(double value);**

**// Сводка:**

**// Удаляет значение с указанным ключом из словаря**

**// Параметры:**

**// key:**

**// Ключ элемента, который необходимо удалить.**

**// Возвращает:**

**// true, если элемент успешно найден и удален, в противном случае — false. Этот**

**// метод возвращает значение false, если ключ key не найден в словаре**

**public bool Remove(string key);**

**// Сводка:**

**// Получает значение, связанное с указанным ключом.**

**// Параметры:**

**// key:**

**// Ключ значения, которое необходимо получить.**

**// value:**

**// Возвращаемое значение, связанное с указанном ключом, если он найден; в противном случае — значение по умолчанию для данного типа параметра value. Этот параметр**

**передается неинициализированным.**

**public bool TryGetValue(string key, out double value);**

**}**