Зонова Анна, ИУ5-34Б

Как реализовать разреженную матрицу на основе словаря в языке C#?

Разреженную матрицу можно реализовать «поверх» стандартной коллекции. В приведенном ниже примере разреженная матрица реализована на основе словаря. Ключом элемента словаря является комбинация индексов ячейки матрицы по строке и столбцу, значением элемента словаря – значение элемента матрицы. Класс «Разреженная матрица» реализован в виде обобщенной коллекции, класс-обобщение T соответствует типу ячейки матрицы.

Пример реализации класса «Разреженная матрица»:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace FigureCollections
{
	public class Matrix <T>
	{
		///<summary>
		///Словарь для хранения значений
		///</summary>
		Dictionary<string, T> _matrix = new Dictionary<string, T>();

		///<summary>
		///Количество элементов по горизонтали (максимальное
	количество столбцов)
		///</summary>
		int maxX;

		///<summary>
		///Количество элементов по вертикали (максимальное
	количество строк)
		///</summary>
		int maxY;

		///<summary>
		///Реализация интерфейса для проверки пустого элемента
		///</summary>
		IMatrixCheckEmpty<T> checkEmpty;

		///<summary>
		///Конструктор
```

```
        ///</summary>
        public Matrix (int px, int py,
                        IMatrixCheckEmpty <T>
    checkEmptyParam)
        {
            this.maxX = px;
            this.maxY=py;
            this.checkEmpty = checkEmptyParam;
        }

        ///<summary>
        ///Индексатор для доступа данных
        ///</summary>
        public T this[int x,  int y]
        {
            set
            {
                CheckBounds(x, y);
                string key = DictKey(x, y);
                this._matrix.Add(key, value);
            }
            get
            {
                CheckBounds(x, y);
                string key = DictKey(x, y);
                if (this._matrix.ContainsKey(key))
                {
                    return this._matrix[key];
                }
                else
                {
                    return this.checkEmpty.getEmptyElement();
                }
            }
        }

///<summary>
///Проверка границ
///</summary>
void CheckBounds (int x, int y)
{
    if (x<0 || x>=this.maxX)
    {
        throw new ArgumentOutRangeException("x", "x=" + x +"выходит
за границы");
```

```
        }
        if (y<0 || y>=this.maxY)
        {
                throw new ArgumentOutRangeException(“y”, “y=” + y +”выходит
        за границы”);
        }
}

///<summary>
///Формирование ключа
///</summary>
string DictKey (int x, int y)
{
        return x.ToString() + “_“+y.ToString();
}

///<summary>
///Приведение к строке
///</summary>
public override string ToString()
{
        StringBuilder b = new StringBuilder();
        for (int j=0; j<this.maxY; j++)
        {
                b.Append(“[”);
                for (int i=0; i<this.maxX; i++)
                {
                        //Добавление разделителя-табуляции
                        if (i>0)
                        {
                                b.Append(‘\t”);
                        }
                        //Если текущий элемент не пустой
                if (!this.checkEmpty.checkEmptyElement(this[I, j]))
                {
                        //Добавить приведённой строке текущий элемент
                                b.Append(this[I, j].ToString());
                }
                else
                 {
                        //Иначе добавить признак пустого значения
                        b.Append(“ . ”);
                }
        }
        b.Append(“]\n”);
```

```
}
return b.ToString();
}
}
}
```

Основная структура данных для хранения разреженной матрицы – словарь _matrix. Ключ словаря – строка, которая содержит комбинацию индексов ячейки матрицы по строке и столбцу, значение словаря – обобщенный тип T, который является типом значения элемента матрицы. Поля данных maxX и maxY используются для хранения размеров матрицы. Поле checkEmpty содержит объект класса, реализующего интерфейс IMatrixCheckEmpty для работы с пустыми значениями.

Класс разреженной матрицы Matrix. Он содержит конструктор, в который в качестве параметров передаются размеры матрицы по горизонтали и вертикали и объект класса, реализующего интерфейс IMatrixCheckEmpty. Конструктор сохраняет параметры в соответствующих полях класса. Наиболее важной частью реализации является индексатор «public T this[int x, int y]». В качестве параметров индексатору передаются координаты x и y – столбец и строка текущей ячейки матрицы. Set-аксессор индексатора выполняет запись элемента в матрицу. Вначале выполняется проверка границ матрицы с помощью метода CheckBounds. Данный метод проверяет, что координаты текущей ячейки x и y находятся в диапазоне от 0 до границы по x или y соответственно. В случае выхода за границы генерируется исключение ArgumentOutOfRangeException с соответствующим сообщением. Если координаты текущей ячейки находятся в пределах требуемых границ, то вычисляется ключ для записи в словарь с помощью функции DictKey. Данная функция осуществляет строковую конкатенацию координат x и y ячейки. Функция DictKey является устойчивой – всегда возвращает одинаковый результат для одного и того же набора параметров. Поэтому данная функция может быть использована как для записи, так и для чтения данных. После вычисления ключа проводится запись значения, присваиваемого в Set-аксессоре (value), в словарь _matrix. В данной реализации не проверяется, существует ли уже в словаре ячейка с таким ключом, как и в случае обычного массива, элементы перезаписываются. Однако, если необходимо, то можно добавить дополнительную проверку на наличие элемента в словаре и запретить перезапись элементов, поскольку при попытке перезаписи может генерироваться исключение. Get-аксессор индексатора осуществляет чтение элемента из матрицы. Как и в случае записи, при чтении сначала проверяются границы матрицы с помощью метода CheckBounds, и если координаты текущей ячейки находятся в пределах требуемых границ, то вычисляется ключ для чтения из словаря посредством функции DictKey. Далее проверяется существование ключа в

словаре с помощью метода _matrix.ContainsKey. Если элемент с вычисленным ключом существует, то Get-аксессор возвращает значение элемента. Если элемент не существует, то возвращается пустое значение с использованием поля checkEmpty. В классе разреженной матрицы также переопределен метод ToString, выводящий значение матрицы в строковом представлении. Алгоритм работы данного метода является обычным, в нем с использованием двух вложенных циклов выполняется перебор и вывод в строку ячеек матрицы. Для работы со строками в данном методе используется класс StringBuilder.