

Министерство науки и высшего образования Российской Федерации
Муромский институт (филиал)
федерального государственного бюджетного образовательного учреждения высшего образования
«Владимирский государственный университет
Имени Александра Григорьевича и Николая Григорьевича Столетовых»
(МИВлГУ)

Факультет _____ ИТ _____

Кафедра _____ ПИН _____

ЛАБОРАТОРНАЯ РАБОТА №2

по _____ САОД _____

Тема _____ Использование коллекций Стекл, Очередь _____

Руководитель

Привезенцев Д.Г. _____
(фамилия, инициалы)

(подпись) _____ (дата)

Студент ПИН-119 _____
(группа)

Лямина И.А. _____
(фамилия, инициалы)

(подпись) _____ (дата)

Лабораторная работа №2

Тема: Использование коллекций Стек, Очередь.

Задание на лабораторную работу

Стек

Список задач по моделированию стека:

1. Реализовать класс-оболочку, реализующую алгоритмы работа стека.
2. Реализовать метод добавления элемента в стек. Метод получает в качестве аргументов добавляемый элемент;
3. Реализовать метод извлечения элемента из стека. Функция возвращает извлекаемое число;
4. Реализовать метод реверсирования (переворачивания) стека. Элементы извлекаются из стека и добавляются в новый до тех пор, пока исходный стек не окажется пустым. Далее необходимо перенаправить указатель со старого стека на новый. Для извлечения и добавления использовать подготовленные ранее методы. Пользоваться связями линейного списка, на базе которого построен стек, запрещается;
5. Реализовать печать стека через двойное реверсирование. Реверсирование производится дважды: первый раз так же, как описано выше (лучше просто вызвать соответствующую функцию), второй раз добавляется печать извлеченного значения.
6. Решить индивидуальную задачу с использованием разработанного выше класса.

					МИВУ.09.03.04.08-02									
Из	Лис	№ докум.	Подп.	Дата	Использование коллекций Стек, Очередь					Лит.		Лист	Листов	
Разраб.	Лямина И.А.			21.04.									2	15
Провер.	Привезенцев Д.Г.									МИ ВлГУ ПИН - 119				
Н.контр.														
Утв.														

Задание 1.

1. Сформировать стек из 10 чисел. Найти сумму последних 5 чисел и результат поместить в стек.
2. Сформировать стек из 8 чисел. Найти среднее арифметическое второго и третьего чисел. Результат поместить в стек.

Задание 2.

Дан стек, заполненный случайным образом из целых чисел. Поменять в данном стеке содержимое вершины и дна.

Задание 3.

Стек заполнен элементами типа *struct* с полем *year* типа *int*. Записать в стек сначала четные элементы, затем – нечетные. Для решения задачи используйте два дополнительных стека.

Выполнение задания по стеку:

Листинг программы:

```
using System;

namespace LB_2
{
    class Program
    {
        public class Node<T>
        {
            public Node(T data)
            {
                Data = data;
            }
            public T Data { get; set; }
            public Node<T> Next { get; set; }
        }

        public class NodeStack<T>
        {
            Node<T> head;
            int count;

            public bool IsEmpty
            {
                get { return count == 0; }
            }
            public int Count
            {
                get { return count; }
            }

            public void Push(T item)
            {

```

```

        // увеличиваем стек
        Node<T> node = new Node<T>(item);
        node.Next = head; // переустанавливаем верхушку стека на новый элемент
        head = node;
        count++;
    }
    public T Pop()
    {
        // если стек пуст, выбрасываем исключение
        if (IsEmpty)
            throw new InvalidOperationException("Стек пуст");
        Node<T> temp = head;
        head = head.Next; // переустанавливаем верхушку стека на следующий элемент
        count--;
        return temp.Data;
    }

    public static void Reverse(ref NodeStack<T> stack)
    {
        var newStack = new NodeStack<T>();
        while (!stack.IsEmpty)
        {
            newStack.Push(stack.Pop());
        }
        stack = newStack;
    }

    public static void Print(ref NodeStack<T> stack)
    {
        var tempStack = new NodeStack<T>();
        Copy(ref stack, ref tempStack);
        Reverse(ref tempStack);
        while (!tempStack.IsEmpty)
        {
            Console.WriteLine(tempStack.Pop());
        }
    }
    public static void Copy(ref NodeStack<T> which, ref NodeStack<T> where)
    {
        var newStack = new NodeStack<T>();
        while (!which.IsEmpty)
        {
            var temp = which.Pop();
            newStack.Push(temp);
            where.Push(temp);
        }
        Reverse(ref newStack);
        Reverse(ref where);
        which = newStack;
    }
}

static void Main(string[] args)
{
    var rnd = new Random();

    Console.WriteLine("1 задание");
    Console.WriteLine("1 пункт\n");
    //1.Сформировать стек из 10 чисел.Найти сумму последних 5 чисел и результат поместить в стек.
    var stack_10 = new NodeStack<int>();
    for(var i = 0; i < 10; i++)
    {
        stack_10.Push(rnd.Next(0, 100));
    }
}

```

```

Console.WriteLine("Изначальный стек");
NodeStack<int>.Print(ref stack_10);
Console.WriteLine("-----");

var tempStack = new NodeStack<int>();
NodeStack<int>.Copy(ref stack_10, ref tempStack);
int sum = 0;
for (var i = 0; i < 5; i++)
{
    sum += tempStack.Pop();
}
stack_10.Push(sum);

Console.WriteLine("Стек после изменения");
NodeStack<int>.Print(ref stack_10);

Console.WriteLine();
Console.WriteLine("2 пункт\n");
//2.Сформировать стек из 8 чисел.Найти среднее арифметическое второго и третьего чисел.Результат
поместить в стек.
var stack_8 = new NodeStack<int>();
for (var i = 0; i < 8; i++)
{
    stack_8.Push(rnd.Next(0, 100));
}
Console.WriteLine("Изначальный стек");
NodeStack<int>.Print(ref stack_8);
Console.WriteLine("-----");

NodeStack<int>.Copy(ref stack_8, ref tempStack);
NodeStack<int>.Reverse(ref tempStack);
sum = 0;
for (var i = 0; i < 3; i++)
{
    var temp = tempStack.Pop();
    if (i != 0)
    {
        sum += temp;
    }
}
stack_8.Push(sum / 2);

Console.WriteLine("Стек после изменения");
NodeStack<int>.Print(ref stack_8);

Console.WriteLine();
Console.WriteLine("2 задание\n");
//Дан стек, заполненный случайным образом из целых чисел.Поменять в данном стеке содержимое вершины и
дна.
var stack_2 = new NodeStack<int>();
for (var i = 0; i < 5; i++)
{
    stack_2.Push(rnd.Next(0, 100));
}
Console.WriteLine("Изначальный стек");
NodeStack<int>.Print(ref stack_2);
Console.WriteLine("-----");

var end = stack_2.Pop();
NodeStack<int>.Reverse(ref stack_2);
var start = stack_2.Pop();
stack_2.Push(end);
NodeStack<int>.Reverse(ref stack_2);
stack_2.Push(start);

```

```

Console.WriteLine("Стек после изменения");
NodeStack<int>.Print(ref stack_2);

Console.WriteLine();
Console.WriteLine("3 задание\n");
//Стек заполнен элементами типа struct с полем year типа int. Записать в стек сначала четные элементы, затем
– нечетные. Для решения задачи используйте два дополнительных стека.
var stack_3 = new NodeStack<task_3>();
task_3 num1; num1.year = 2000; stack_3.Push(num1);
task_3 num2; num2.year = 2001; stack_3.Push(num2);
task_3 num3; num3.year = 2002; stack_3.Push(num3);
task_3 num4; num4.year = 2003; stack_3.Push(num4);
task_3 num5; num5.year = 2004; stack_3.Push(num5);
task_3 num6; num6.year = 2005; stack_3.Push(num6);
task_3 num7; num7.year = 2006; stack_3.Push(num7);
Console.WriteLine("Изначальный стек");
NodeStack<task_3>.Print(ref stack_3);
Console.WriteLine("-----");

var stack_even = new NodeStack<task_3>();
var stack_uneven = new NodeStack<task_3>();
while (!stack_3.IsEmpty)
{
    var temp = stack_3.Pop();
    if (temp.year % 2 == 0) stack_even.Push(temp);
    else stack_uneven.Push(temp);
}
while (!stack_even.IsEmpty)
{
    stack_3.Push(stack_even.Pop());
}
while (!stack_uneven.IsEmpty)
{
    stack_3.Push(stack_uneven.Pop());
}

Console.WriteLine("Стек после изменения");
NodeStack<task_3>.Print(ref stack_3);

Console.ReadLine();
}
struct task_3
{
    public int year;
    public override string ToString()
    {
        return year.ToString();
    }
}
}

```

```

1 задание
1 пункт

Изначальный стек
9
91
98
33
81
4
38
33
38
88
-----
Стек после изменения
9
91
98
33
81
4
38
33
38
88
201

2 пункт

Изначальный стек
11
44
3
0
10
60
49
31
-----
Стек после изменения
11
44
3
0
10
60
49
31
23

```

Рисунок 1 – скриншот работы 1 задания (стек)

					МИВУ.09.03.04.08-02	Лист
						7
Из	Лист	№ докум.	Подп.	Дата		

```

2 задание

Изначальный стек
42
38
50
46
28
-----
Стек после изменения
28
38
50
46
42

3 задание

Изначальный стек
2000
2001
2002
2003
2004
2005
2006
-----
Стек после изменения
2000
2002
2004
2006
2001
2003
2005

```

Рисунок 2 – скриншот работы 2 и 3 задания (стек)

Очередь

Задание 1.

Заполнить очередь случайными целыми числами и отсортировать её по убыванию.

Задание 2

Дана очередь из целых чисел, заданных случайным образом. Оставить в очереди только двухзначные числа, все остальные удалить.

Задание 3

Дана очередь из целых чисел. Удалить из нее числа кратные заданному с клавиатуры числу.

					МИВУ.09.03.04.08-02	Лист
						8
Из	Лист	№ докум.	Подп.	Дата		

Выполнение задания по очереди:

Листинг программы:

```
using System;
using System.Collections.Generic;
using System.Linq;

namespace LB_2_1
{
    class Program
    {
        static void Main(string[] args)
        {
            var rnd = new Random();

            Console.WriteLine("1 задание\n");
            //Заполнить очередь случайными целыми числами и отсортировать её по убыванию.
            var queue_1 = new Queue<int>();
            for (var i = 0; i < 10; i++)
            {
                queue_1.Enqueue(rnd.Next(0, 100));
            }
            Console.WriteLine("Изначальная очередь");
            foreach (var item in queue_1)
            {
                Console.WriteLine(item);
            }
            Console.WriteLine("-----");

            Queue<int> queue_sort = new Queue<int>(queue_1.OrderByDescending(z => z));

            Console.WriteLine("Очередь после изменения");
            foreach (var item in queue_sort)
            {
                Console.WriteLine(item);
            }

            Console.WriteLine();
            Console.WriteLine("2 задание\n");
            //Дана очередь из целых чисел, заданных случайным образом. Оставить в очереди только двухзначные числа,
            //все остальные удалить.
            var queue_2 = new Queue<int>();
            for (var i = 0; i < 10; i++)
            {
                queue_2.Enqueue(rnd.Next(0, 150));
            }
            Console.WriteLine("Изначальная очередь");
            foreach (var item in queue_2)
            {
                Console.WriteLine(item);
            }
            Console.WriteLine("-----");

            var queue_result = new Queue<int>();
            while(queue_2.Count != 0)
            {
                var temp = queue_2.Dequeue();
                if(temp / 100 == 0 && temp / 10 != 0)
                {
                    queue_result.Enqueue(temp);
                }
            }
        }
    }
}
```

					МИВУ.09.03.04.08-02	Лист
						9
Из	Лист	№ докум.	Подп.	Дата		

```

Console.WriteLine("Очередь после изменения");
foreach (var item in queue_result)
{
    Console.WriteLine(item);
}

Console.WriteLine();
Console.WriteLine("3 задание\n");
//Дана очередь из целых чисел. Удалить из нее числа кратные заданному с клавиатуры числу.
Console.Write("Введите число = ");
int number = Convert.ToInt32(Console.ReadLine());
var queue_3 = new Queue<int>();
for (var i = 0; i < 10; i++)
{
    queue_3.Enqueue(rnd.Next(0, 100));
}
Console.WriteLine("Изначальная очередь");
foreach (var item in queue_3)
{
    Console.WriteLine(item);
}
Console.WriteLine("-----");

var queue_res = new Queue<int>();
while (queue_3.Count != 0)
{
    var temp = queue_3.Dequeue();
    if (temp % number != 0)
    {
        queue_res.Enqueue(temp);
    }
}

Console.WriteLine("Очередь после изменения");
foreach (var item in queue_res)
{
    Console.WriteLine(item);
}
}
}

```

1 задание

Изначальная очередь

55

31

6

83

4

11

94

54

35

89

Очередь после изменения

94

89

83

55

54

35

31

11

6

4

2 задание

Изначальная очередь

5

80

56

87

65

6

148

147

113

94

Очередь после изменения

80

56

87

65

94

Рисунок 3 – скриншот работы 1 и 2 задания (очередь)

					МИВУ.09.03.04.08-02	Лист
						11
Из	Лист	№ докум.	Подп.	Дата		

```

3 задание

Введите число = 2
Изначальная очередь
78
91
37
13
2
17
28
80
3
16
-----
Очередь после изменения
91
37
13
17
3

```

Рисунок 4 – скриншот работы 3 задания (очередь)

Дек

Найти сумму всех элементов дека, который состоит из целых чисел, и приписать эту сумму в начало дека.

Выполнение задания по деку:

Листинг программы:

```

using System;
using System.Collections;
using System.Collections.Generic;

namespace LB_2_2
{
    class Program
    {
        public class DoublyNode<T>
        {
            public DoublyNode(T data)
            {
                Data = data;
            }
            public T Data { get; set; }
            public DoublyNode<T> Previous { get; set; }
            public DoublyNode<T> Next { get; set; }
        }
    }
}

```

					МИВУ.09.03.04.08-02	Лист
						12
Из	Лист	№ докум.	Подп.	Дата		

```

public class Deque<T> : IEnumerable<T> // двусвязный список
{
    DoublyNode<T> head; // головной/первый элемент
    DoublyNode<T> tail; // последний/хвостовой элемент
    int count; // количество элементов в списке

    // добавление элемента
    public void AddLast(T data)
    {
        DoublyNode<T> node = new DoublyNode<T>(data);

        if (head == null)
            head = node;
        else
        {
            tail.Next = node;
            node.Previous = tail;
        }
        tail = node;
        count++;
    }
    public void AddFirst(T data)
    {
        DoublyNode<T> node = new DoublyNode<T>(data);
        DoublyNode<T> temp = head;
        node.Next = temp;
        head = node;
        if (count == 0)
            tail = head;
        else
            temp.Previous = node;
        count++;
    }
    public T RemoveFirst()
    {
        if (count == 0)
            throw new InvalidOperationException();
        T output = head.Data;
        if (count == 1)
        {
            head = tail = null;
        }
        else
        {
            head = head.Next;
            head.Previous = null;
        }
        count--;
        return output;
    }
    public T RemoveLast()
    {
        if (count == 0)
            throw new InvalidOperationException();
        T output = tail.Data;
        if (count == 1)
        {
            head = tail = null;
        }
        else
        {
            tail = tail.Previous;
            tail.Next = null;
        }
        count--;
    }
}

```

```

        return output;
    }
    public int Count { get { return count; } }
    public bool IsEmpty { get { return count == 0; } }

    IEnumerator IEnumerable.GetEnumerator()
    {
        return ((IEnumerable)this).GetEnumerator();
    }

    IEnumerator<T> IEnumerable<T>.GetEnumerator()
    {
        DoublyNode<T> current = head;
        while (current != null)
        {
            yield return current.Data;
            current = current.Next;
        }
    }
}

static void Main(string[] args)
{
    var rnd = new Random();
    //Найти сумму всех элементов дека, который состоит из целых чисел, и приписать эту сумму в начало дека.
    var deque = new Deque<int>();
    for (var i = 0; i < 5; i++)
    {
        deque.AddLast(rnd.Next(0, 10));
    }
    Console.WriteLine("Изначальный дек");
    foreach (var item in deque)
    {
        Console.WriteLine(item);
    }
    Console.WriteLine("-----");

    var deque_result = new Deque<int>();
    int sum = 0;
    while (!deque.IsEmpty)
    {
        var temp = deque.RemoveFirst();
        deque_result.AddLast(temp);
        sum += temp;
    }
    deque_result.AddFirst(sum);

    Console.WriteLine("Дек после изменения");
    foreach (var item in deque_result)
    {
        Console.WriteLine(item);
    }
}
}

```

```
Консоль отладки Microsoft Visual Stuc
Изначальный дек
5
6
8
6
2
-----
Дек после изменения
27
5
6
8
6
2
```

Рисунок 5 – скриншот работы программы (дек)

Вывод: в ходе данной лабораторной работы были приобретены навыки по работе со стеком и очередью.