

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»

Факультет _____ ИТР _____

Кафедра _____ ПИН _____

ЛАБОРАТОРНАЯ РАБОТА №2

По Структуры и алгоритмы обработки данных

Руководитель

Привезенцев Д.Г.

(фамилия, инициалы)

(подпись)

(дата)

Студент ПИН - 121

(группа)

Ермилов М.В.

(фамилия, инициалы)

(подпись)

(дата)

Муром 2023

Лабораторная работа №2

Тема: Использование коллекций `Стек`, `Очередь`

Ход работы:

Задания для класса `Stack`

- 1) 1. Сформировать стек из 8 чисел. Найти произведение второго и третьего чисел. Результат поместить в стек.
2. Сформировать стек из 7 чисел. Найти сумму 3-го, 4-го и 5-го чисел из стека. Результат поместить в стек.
- 2) Удалить из стека, который составлен из целых чисел, заданных случайным образом, каждый второй элемент. На дне находится первый элемент.
- 3) Дан стек, заполненный элементами типа `struct`. Удалить из стека предпоследний элемент.

Задания для `Queue`

- 1) Заполнить очередь случайным образом целыми числами. Найти максимальный элемент в образовавшейся очереди и вставить до и после него ноль.
- 2) Очередь состоит из целых отрицательных и положительных чисел, заданных случайным образом. Найти и записать вместо положительных элементов, равные им по модулю отрицательные числа.
- 3) Очередь заполнена случайным образом целыми числами. Добавить в начало очереди произведение всех элементов.

Задания для `Deque`

- 1) Заполнить дек случайным образом целыми числами. Найти максимальный элемент в образовавшемся deque и вставить до и после него ноль.

					МИ ВлГУ 09.03.04			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.		Ермилов М.В.						
Провер.		Привезенцев Д.Г.						
Реценз.								
Н. Контр.								
Утверд.								
						Лит.	Лист	Листов
							2	8
						ПИН-121		

Код по заданиям:

```
using Nito.Collections;
using System.Collections;
using System.Collections.Generic;

class program
{
    static void Main(string[] arg)
    {
        Console.WriteLine("Работа с Stack");
        StackTask1();
        StackTask2(10);
        StackTask3(3);
        Console.WriteLine("\n\nРабота с Queue");
        QueueTask1(10);
        QueueTask2(10);
        QueueTask3(10);
        Console.WriteLine("\n\nРабота с Deque");
        DequeTask1(10);
    }

    static int Get(Stack<int> stack, int i)
    {
        if(i < stack.Count)
        {
            Stack<int> N = new Stack<int>();
            int r = 0;
            for (int I = 0; I <= i; I++)
            {
                r = stack.Pop();
                N.Push(r);
            }
            for (int I = 0; I <= i; I++)
            {
                int b = N.Pop();
                stack.Push(b);
            }
            return r;
        }
        throw new Exception();
    }
    static int Get(Queue<int> stack, int i)
    {
        if (i < stack.Count)
        {
            int r = 0;
            for (int I = 0; I < stack.Count; I++)
            {
                int b = stack.Dequeue();
                if (i == I)
                    r = b;
                stack.Enqueue(b);
            }
            return r;
        }
        throw new Exception();
    }
    static int Get(Deque<int> stack, int i)
    {
        if(i < stack.Count)
```

					МИ ВлГУ 09.03.04	Лист
Изм.	Лист	№ докум.	Подпись	Дата		3

```

    {
        int r = 0;
        for(int I = 0; I < stack.Count; I++)
        {
            int b = stack.RemoveFromFront();
            if (i == I)
                r = b;
            stack.AddToBack(b);
        }
        return r;
    }

    throw new Exception();
}

static void StackTask1()
{
    Console.WriteLine("Задание 1");
    Console.WriteLine("Задание 1.1");
    Stack<int> stack = new Stack<int>();
    for(int i = 0; i < 8; i++)
    {
        stack.Push(Rand());
    }
    Console.WriteLine(String(stack));

    //stack.Push(stack.ElementAt(2) + stack.ElementAt(3));
    stack.Push(Get(stack, 2) + Get(stack, 3));
    Console.WriteLine(String(stack));

    Console.WriteLine("Задание 1.2");
    stack = new Stack<int>();
    for (int i = 0; i < 7; i++)
    {
        stack.Push(Rand());
    }
    Console.WriteLine(String(stack));
    //stack.Push(stack.ElementAt(3) + stack.ElementAt(4) + stack.ElementAt(5));
    stack.Push(Get(stack, 3) + Get(stack, 4) + Get(stack, 5));
    Console.WriteLine(String(stack));
}

static void StackTask2(int count)
{
    Console.WriteLine("Задание 2");
    Stack<int> stack = new Stack<int>();
    for (int i = 0; i < count; i++)
    {
        stack.Push(Rand());
    }
    Console.WriteLine(String(stack));

    Stack<int> _stack = new Stack<int>();
    for(int i = 1; i < count; i += 2)
    {
        _stack.Push(Get(stack, count - i));
    }
    stack = _stack;
    Console.WriteLine("Преобразование по заданию:");
    Console.WriteLine(String(stack));
}

static void StackTask3(int count)
{
    Console.WriteLine("Задание 3");
    Stack<forTask3> stack = new Stack<forTask3>();
    for (int i = 0; i < count; i++)

```

```

    {
        stack.Push(new forTask3(Rand(), Rand()));
    }
    Console.WriteLine(String(stack));
    forTask3 a = stack.Pop();
    stack.Pop();
    stack.Push(a);
    Console.WriteLine("Преобразование по заданию:");
    Console.WriteLine(String(stack));
}

static void QueueTask1(int count)
{
    Console.WriteLine("Задание 1");
    Queue<int> queue = new Queue<int>();
    for(int i = 0; i < count; i++)
    {
        queue.Enqueue(Rand());
    }

    Console.WriteLine(String(queue));
    int max = Get(queue, 0);
    for(int i = 1; i < count; i++)
    {
        int a = Get(queue, i);
        if (a > max)
        {
            max = a;
        }
    }
    Queue<int> _queue = new Queue<int>();
    for (int i = 0; i < count; i++)
    {
        int a = Get(queue, i);
        if (i != 0)
        {
            if(Get(queue, i - 1) == max)
            {
                a = 0;
            }
        }
        if (i < count - 1)
        {
            if (Get(queue, i + 1) == max)
            {
                a = 0;
            }
        }
        _queue.Enqueue(a);
    }
    queue = _queue;
    Console.WriteLine("Преобразование по заданию:");
    Console.WriteLine(String(queue));
}

static void QueueTask2(int count)
{
    Console.WriteLine("Задание 2");
    Queue<int> queue = new Queue<int>();
    for (int i = 0; i < count; i++)
    {
        queue.Enqueue(Rand(-100, 100));
    }
    Console.WriteLine(String(queue));

    Queue<int> _queue = new Queue<int>();
    for (int i = 0; i < count; i++)

```

```

    {
        if (Get(queue, i) > 0)
            _queue.Enqueue(Get(queue, i) * -1);
        else
            _queue.Enqueue(Get(queue, i));
    }
    queue = _queue;

    Console.WriteLine("Преобразование по заданию:");
    Console.WriteLine(String(queue));
}
static void QueueTask3(int count)
{
    Console.WriteLine("Задание 3");
    Queue<int> queue = new Queue<int>();
    for (int i = 0; i < count; i++)
    {
        queue.Enqueue(Rand(1, 100));
    }
    Console.WriteLine(String(queue));

    Queue<int> _queue = new Queue<int>();
    int a = 1;
    for (int i = 0; i < count; i++)
    {
        a *= Get(queue, i);
    }
    _queue.Enqueue(a);
    for (int i = 0; i < count; i++)
    {
        _queue.Enqueue(Get(queue, i));
    }
    queue = _queue;

    Console.WriteLine("Преобразование по заданию:");
    Console.WriteLine(String(queue));
}

static void DequeueTask1(int count)
{
    Console.WriteLine("Задание 1");
    Deque<int> deque = new Deque<int>();
    for(int i = 0; i < count; i++)
    {
        deque.AddToBack(Rand());
    }
    int max = Get(deque, 0);
    for(int i = 1; i < count; i++)
    {
        if (Get(deque, i) > max)
            max = Get(deque, i);
    }
    Console.WriteLine(String(deque));

    Deque<int> _deque = new Deque<int>();
    for (int i = 0; i < count; i++)
    {
        int a = Get(deque, i);
        if (i != 0)
        {
            if (Get(deque, i - 1) == max)
                a = 0;
        }
        if(i < count - 1)
        {

```

```

        if (Get(deque, i + 1) == max)
            a = 0;
    }
    _deque.AddToBack(a);
}
deque = _deque;
Console.WriteLine("Преобразование по заданию:");
Console.WriteLine(String(deque));
}

static string String(Stack<int> stack)
{
    string r = "";
    for(int i = 0; i < stack.Count; i++)
    {
        r += stack.ElementAt(i) + " ";
    }
    return r;
}
static string String(Queue<int> queue)
{
    string r = "";
    for (int i = 0; i < queue.Count; i++)
    {
        r += queue.ElementAt(i) + " ";
    }
    return r;
}
static string String(Stack<forTask3> stack)
{
    string r = "";
    for (int i = 0; i < stack.Count; i++)
    {
        r += stack.ElementAt(i) + " ";
    }
    return r;
}
static string String(Deque<int> deque)
{
    string r = "";
    for (int i = 0; i < deque.Count; i++)
    {
        r += deque.ElementAt(i) + " ";
    }
    return r;
}
static Random rnd = new Random();
static int Rand() => rnd.Next(0, 100);
static int Rand(int min, int max) => rnd.Next(min, max);
struct forTask3
{
    public int x;
    public int y;
    public forTask3(int x, int y)
    {
        this.x = x;
        this.y = y;
    }
    public override string ToString() => $"(x:{x} y:{y})";
}
}

```

```

Работа с Stack
Задание 1
Задание 1.1
78 26 55 33 61 26 88 17
88 78 26 55 33 61 26 88 17
Задание 1.2
72 4 92 56 40 59 58
155 72 4 92 56 40 59 58
Задание 2
99 98 61 5 25 0 78 55 27 24
Преобразование по заданию:
98 5 0 55 24
Задание 3
(x:84 y:76) (x:31 y:45) (x:29 y:2)
Преобразование по заданию:
(x:84 y:76) (x:29 y:2)

Работа с Queue
Задание 1
36 44 82 76 56 10 3 10 86 11
Преобразование по заданию:
36 44 82 76 56 10 3 0 86 0
Задание 2
-77 -59 59 70 -14 -97 16 90 -6 5
Преобразование по заданию:
-77 -59 -59 -70 -14 -97 -16 -90 -6 -5
Задание 3
60 89 35 33 66 30 67 64 64 23
Преобразование по заданию:
-726204416 60 89 35 33 66 30 67 64 64 23

Работа с Deque
Задание 1
9 26 37 97 20 48 50 8 53 27
Преобразование по заданию:
9 26 0 97 0 48 50 8 53 27

```

Рис 1 - пример работы кода.