

Учреждение образования "Полоцкий Государственный Университет"

Факультет информационных технологий
Кафедра вычислительных систем и сетей

Отчет по Лабораторной работе № 4

по дисциплине: **"Объектно-Оrientированные Технологии
Программирования и Стандарты Проектирования"**

ВЫПОЛНИЛ

студент группы 18-IT-2
Сыцевич Д.Н.
вариант № 1

ПРОВЕРИЛ

старший преподаватель
Ярошевич П.В.

Полоцк 2020 г.

1 Задача

Создать обобщенный класс, указанной в варианте задания, структуры данных и реализовать дополнительные функции для неё. Структура данных должна быть реализованна на основе массива или ссылок

(при реализации нельзя использовать никакие другие коллекции).

2 Вариант № 1

Структура данных “Стек”. Дополнительно реализовать функцию проверки структур на равенство.

3 Ход выполнения

Выполнение лабораторной работы включало в себя следующие шаги:

1. Создание класса Stack.
2. Создание класса Program.
3. Компиляция и запуск программы.
4. Написание отчёта.
5. Подготовка к защите.

4 Скриншоты

```
Ссылка: 13
class MyStack<T> where T:Comparable
{
    Ссылка: 16
    private class Data
    {
        private Data next;
        private T value;

        Ссылка: 14
        public Data getNext()...
        Ссылка: 3
        public void setNext(Data next)...
        Ссылка: 6
        public T getValue()...
        Ссылка: 3
        public void setValue(T value)...

        Ссылка: 2
        public Data()...

        Ссылка: 1
        public String toString()...
    };

    private Data head;
    private Data start;

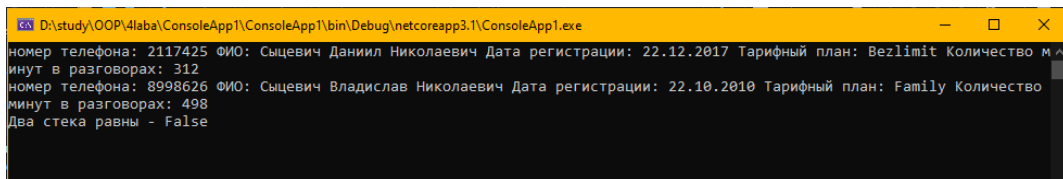
    Ссылка: 4
    public MyStack()...

    Ссылка: 3
    public void push(T obj)...

    Ссылка: 2
    public T pop()...
    Ссылка: 0
    public void clear()
    {
        while (head != start)
        {
            Data iterator = start;
            while (iterator.getNext() != null && iterator.getNext() != head)
            {
                iterator = iterator.getNext();
            }
            head = iterator;
            head.setNext(null);
        }
    }

    Ссылка: 0
    public void print()
}
```

Рис. 1: Скриншот класса Stack



```
D:\study\OOP\4lab\\ConsoleApp1\ConsoleApp1\bin\Debug\netcoreapp3.1\ConsoleApp1.exe
номер телефона: 2117425 ФИО: Сыцевич Даниил Николаевич Дата регистрации: 22.12.2017 Тарифный план: Bezlimit Количество м ^
инут в разговорах: 312
номер телефона: 8998626 ФИО: Сыцевич Владислав Николаевич Дата регистрации: 22.10.2010 Тарифный план: Family Количество
минут в разговорах: 498
Два стека равны - False
```

Рис. 2: Скриншот выполнения с примитивными типами данных

5 Source Code

MyStack.cs

```
using System;
using System.Collections.Generic;
using System.Text;

namespace oopl
{
    class MyStack<T> where T:IComparable
    {
        private class Data
        {
            private Data next;
            private T value;

            public Data getNext()
            {
                return next;
            }
            public void setNext(Data next)
            {
                this.next = next;
            }
            public T getValue()
            {
                return value;
            }
            public void setValue(T value)
            {
                this.value = value;
            }

            public Data()
            {
                value = default;
                next = default;
            }
        }
    }
}
```

```

        public String toString()
        {
            return value + "␣";
        }
};

private Data head;
private Data start;

public MyStack()
{
    start = new Data();
    head = start;
}

public void push(T obj)
{
    Data ne = new Data();
    ne.setValue(obj);
    head.setNext(ne);
    head = ne;
}

public T pop()
{
    if (head == start) return default;
    Data oh = head;
    Data iterator = start;
    while (iterator.getNext() != null && iterator
    {
        iterator = iterator.getNext();
    }
    head = iterator;
    head.setNext(null);
    return oh.getValue();
}

public void clear()
{
    while (head != start)

```

```

        {
            Data iterator = start;
            while (iterator.getNext() != null &&
            {
                iterator = iterator.getNext();
            }
            head = iterator;
            head.setNext(null);
        }
    }
    public void print()
    {
        if (head == start) return;
        Data iterator = start.getNext();
        while (iterator != null)
        {
            Console.WriteLine(iterator.toString());
            iterator = iterator.getNext();
        }
    }

    public MyStack<T> clone()
    {
        if (head == start) return new MyStack<T>();
        MyStack<T> cl = new MyStack<T>();
        for (Data iterator = start.getNext(); iterator != null; iterator = iterator.getNext())
        {
            cl.push(iterator.getValue());
        }
        return cl;
    }

    public override bool Equals(object obj)
    {
        if (!(obj is MyStack<T>)) return false;
        MyStack<T> sobj = (MyStack<T>)obj;
        MyStack<T> mcl = clone();
        while (sobj.head != sobj.start || mcl.head != mcl.start)
        {
            if (!sobj.pop().Equals(mcl.pop()))

```



```

protected internal string date; //data (dd.mm.yyyy)
protected internal string tarif; //stroka
protected internal int minut; //chislo
protected internal phone(int nomer, string FIO, string date,
{
    this.nomer = nomer;
    this.FIO = FIO;
    this.date = date;
    this.tarif = tarif;
    this.minut = minut;
}

protected internal phone()
{
    nomer = 2117425;
    FIO = "
    date = "22.12.2017";
    tarif = "Bezlimit";
    minut = 312;
}

protected internal phone(phone copy)
{
    nomer = copy.nomer;
    FIO = copy.FIO;
    date = copy.date;
    tarif = copy.tarif;
    minut = copy.minut;
}

protected void SetNomer(int nomer)
{ this.nomer = nomer; }

protected int GetNomer()
{ return nomer; }

protected void SetFIO(string FIO)
{ this.FIO = FIO; }

protected string GetFIO()

```

```

{ return FIO; }

protected void SetDate(string date)
{ this.date = date; }

protected string GetDate()
{ return date; }

protected void SetTarif(string tarif)
{ this.tarif = tarif; }

protected string GetTarif()
{ return tarif; }

protected void SetMinut(int minut)
{ this.minut = minut; }

protected int GetMinut()
{ return minut; }

public override bool Equals(object obj)
{
    if (obj == null)
        return false;
    phone m = obj as phone; // null
    if (m as phone == null)
        return false;
    return m.nomer == this.nomer && m.FIO == this.FIO && m.d
}

public override string ToString()
{
    return "
+ "      :_ " + nomer
+ "      :_ " + FIO
+ "      :_ " + date
+ "      :_ " + tarif
+ "
"
}

```

```

public int CompareTo(object obj)
{
    if (nomer > ((phone)obj).nomer)
        return 1;
    else if (nomer == ((phone)obj).nomer)
        return 0;
    else
        return -1;
}
}

class Program
{
    static void Main(string[] args)
    {
        phone phone1 = new phone();
        Console.WriteLine(phone1);
        phone phone2 = new phone(8998626, "
        Console.WriteLine(phone2);
        MyStack<phone> stack1 = new MyStack<phone>();
        MyStack<phone> stack2 = new MyStack<phone>();

        for(int i = 0;i<10;i++)
        {
            stack1.push(phone1);
            stack2.push(phone2);
        }
        Console.WriteLine("
        Console.ReadKey();
    }
}
}

```