

**Министерство образования Республики Беларусь
Учреждение образования «Полоцкий государственный
университет им. Евфросинии
Полоцкой»**

Кафедра технологий
программирования

**Конструирование программного обеспечения
Отчет по лабораторной работе №5
Вариант 12**

Выполнил

Ланцев Евгений Николаевич.
21-ИТ-1, ФИТ

Проверил

преподаватель
Дьякова А.С.

Полоцк
2022 г.

Лабораторная работа № 5

“Работа с динамическими структурами данных”

Цель работы: Познакомится с работой динамических структур данных. Разобрать методы для работы с динамическими структурами данных в C++ и C#. На основе примеров, приведенных в данной лабораторной работе, выполнить свой вариант практического задания.

Вариант 12

1 Задание

Написал программу которая переводит выражение из инфиксной форме в префиксную.

```
var exp = "A+B*C+D";
// The main method that converts given infix expression
// to prefix expression.
public static string InfixToPrefix(string exp)
{
    Stack<char> ops = new();

    string result = string.Empty;

    for (int i = 0; i < exp.Length; i++)
    {
        // If char is letter or digit then add to result string
        if (char.IsLetterOrDigit(exp[i]))
            result = exp[i] + result;

        // if char is operator
        if (Operators.ContainsKey(exp[i]))
        {
            // if in stack are ops with higher prior
            while (ops.Count > 0 && Operators[exp[i]] <= Operators[ops.Peek()])
            {
                result = ops.Pop() + result;
            }
            // push operator to stack
            ops.Push(exp[i]);
        }
    }

    while (ops.Count != 0)
    {
        result = ops.Pop() + result;
    }

    return result;
}
```

Результат работы программы:
+D+*CBA

2 Задание

Структура Больницы

```
public struct Hospital
{
    private Queue<Patient> _space;
    private int _location;
    private int _capacity;

    public Hospital(int space, int location)
    {
        _space = new Queue<Patient>();
        _capacity = space;
        _location = location;
    }

    public int Distance(int patientLocation)
    {
        return Math.Abs(_location - patientLocation);
    }

    public bool AddPatient(Patient patient)
    {
        if (_space.Count != _capacity)
        {
            Console.WriteLine($"The patient was admitted to the nearest free
hospital");
            _space.Enqueue(patient);
            return true;
        }

        return false;
    }

    public bool DischargePatient()
    {
        if (_space.Count != 0)
        {
            Console.WriteLine($"The patient {_space.Dequeue().Name} was
discharged");
            return true;
        }
        Console.WriteLine($"The hospital is empty");
        return false;
    }

    public void PrintInfo()
    {
        Console.WriteLine($"Number of seats : \t{_capacity}");
        Console.WriteLine($"Number of available seats : \t{_capacity -
_space.Count}");
        Console.WriteLine("At the moment there are:");
        foreach (var patient in _space)
        {
            Console.WriteLine($" \t{patient.Name};");
        }
    }
}
```

Класс пациента:

```
public class Patient
{
    public string Name;
    private int _location;
    private Hospital? _hospital;

    public Patient(string name, int location, List<Hospital> hospitals)
    {
        Name = name;
        _location = location;

        // start with the closest hospital
        foreach (var hospital in hospitals.OrderBy(x => x.Distance(_location)))
        {
            if (hospital.AddPatient(this))
            {
                _hospital = hospital;
                return;
            }
        }

        if(_hospital is null)
            Console.WriteLine($"All hospitals are full, {Name} is going to die");
    }
}
```

Работа с классом и структурой:

```
List<HospitalTask.Hospital> hospitals = new List<HospitalTask.Hospital>();
hospitals.Add(new HospitalTask.Hospital(10,42));
hospitals.Add(new HospitalTask.Hospital(1,4));
hospitals.Add(new HospitalTask.Hospital(245,152));

var patient1 = new HospitalTask.Patient("Putin",23,hospitals);
var patient2 = new HospitalTask.Patient("Alex",2,hospitals);
var patient3 = new HospitalTask.Patient("Sasha",356,hospitals);
var patient4 = new HospitalTask.Patient("Oleg",2,hospitals);
```

Результат работы программы:

Hospital 1

Number of seats : 10
Number of available seats : 8
At the moment there are:
Vanya;
Oleg;

Hospital 2

Number of seats : 1
Number of available seats : 0
At the moment there are:

Alex;

Hospital 3

Number of seats : 245

Number of available seats : 244

At the moment there are:

Sasha;

The patient Vanya was discharged

Hospital 1 edited

Number of seats : 10

Number of available seats : 9

At the moment there are:

Oleg;

Выводы: Научился использовать динамические структуры данных.