

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №5
по ООПиП

Выполнили:
Студент 2 курса
Группы АС-53
Брашевец Д. Г.
Проверил:
Давидюк Ю.И..

Брест 2020

Вариант 4

АТД - множество с элементами типа char. Дополнительно перегрузить следующие операции:

+ - добавить элемент в множество (типа set+char);

* - пересечение множеств;

int() - мощность множества.

Этот файл содержит функцию "main". Здесь начинается и заканчивается выполнение программы.

//

```
#include "pch.h"
```

```
#include "Set.h"
```

```
int main() {
    Set a, b;
    a.Input();
    b.Input();
    std::cout << "set1: ";
    a.Print();
    std::cout << "set2: ";
    b.Print();
    std::cout << "set1 + 'a': ";
    (a + 'a').Print();
    std::cout << "\nset1 * set2: ";
    (a * b).Print();
    int q = a;
    std::cout << "\nqantity a: " << q;
}
```

// Запуск программы: CTRL+F5 или меню "Отладка" > "Запуск без отладки"

// Отладка программы: F5 или меню "Отладка" > "Запустить отладку"

// Советы по началу работы

// 1. В окне обозревателя решений можно добавлять файлы и управлять ими.

// 2. В окне Team Explorer можно подключиться к системе управления версиями.

// 3. В окне "Выходные данные" можно просматривать выходные данные сборки и другие сообщения.

// 4. В окне "Список ошибок" можно просматривать ошибки.

// 5. Последовательно выберите пункты меню "Проект" > "Добавить новый элемент", чтобы создать файлы кода, или "Проект" > "Добавить существующий элемент", чтобы добавить в проект существующие файлы кода.

// 6. Чтобы снова открыть этот проект позже, выберите пункты меню "Файл" > "Открыть" > "Проект" и выберите SLN-файл.

Файл Set.h

```
#pragma once
```

```
#include<iostream>
```

```

class Set {
private:
    static int MAX_SIZE;
    int aSize;
    char *apArr;
public:
    operator int();
    Set operator + (char);
    Set operator * (const Set&);
    void Print();
    void Input();
    void Add(char);
    bool ItIs(char) const;
    Set();
    Set(const Set&);
    ~Set();
};

int Set::MAX_SIZE = 225;

Set::Set() : apArr(nullptr), aSize(0) {}
Set::Set(const Set &oth) {
    aSize = oth.aSize;
    apArr = new char[aSize];
    for (int i = 0; i < aSize; i++)
        apArr[i] = oth.apArr[i];
}
bool Set::ItIs(char ch) const{
    for (int i = 0; i < aSize; i++)
        if (apArr[i] == ch) return 1;
    return 0;
}
Set::~~Set() {
    delete[] apArr;
}
void Set::Add(char ch) {
    if (ItIs(ch) || aSize == MAX_SIZE) return;
    char *pTmp = new char[aSize + 1];
    for (int i = 0; i < aSize; i++)
        pTmp[i] = apArr[i];
    pTmp[aSize++] = ch;
    delete[] apArr;
    apArr = pTmp;
}
void Set::Input() {
    char ch;
    while (std::cin >> ch && aSize < MAX_SIZE && ch != '.')
        Add(ch);
}
void Set::Print() {

```

```

        for (int i = 0; i < aSize; i++)
            std::cout << apArr[i] << " ";
        std::cout << std::endl;
    }
    Set::operator int() {
        return aSize;
    }
    Set Set::operator+(char ch) {
        Set newSet;
        for (int i = 0; i < aSize; i++)
            newSet.Add(apArr[i]);
        newSet.Add(ch);
        return newSet;
    }
    Set Set::operator*(const Set &oth) {
        Set newSet;
        for (int i = 0; i < aSize; i++)
            for (int j = 0; j < oth.aSize; j++)
                if (apArr[i] == oth.apArr[j])
                    newSet.Add(apArr[i]);
        return newSet;
    }
}

```