Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский государственный технический университет имени Н.Э. Баумана»
(национальный исследовательский университет)
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

О т ч е т

по лабораторной работе № 8

Название лабораторной работы: Программирование с использованием классов в C++. Создание контейнеров.
Дисциплина: Объектно-ориентированное программирование

Студент гр. ИУ6-22Б _____ **А. П. Плютто**
(Подпись, дата) (И.О. Фамилия)

Преподаватель _____ **О. А. Веселовская**
(Подпись, дата) (И.О. Фамилия)

Москва, 2023

Задание

Моделировать множество, в качестве элементов которого могут использоваться числа заданного диапазона и символы (стандартный тип "множество" не использовать). Операции: добавление элемента, удаление элемента, печать элементов и проверка вхождения. Создать класс-потомок, который содержит функцию определения мощности множества. Тестировать полученную модель. Примечание: сначала реализовать и отладить структуру данных как класс, после чего преобразовать класс в шаблон.

В отчете представить диаграмму классов и обосновать выбранную структуру представления данных.

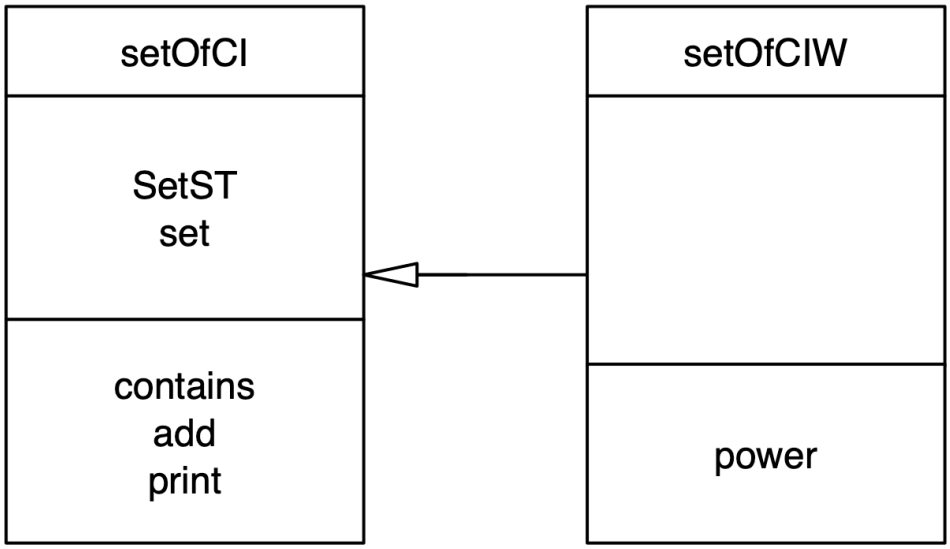
Решение

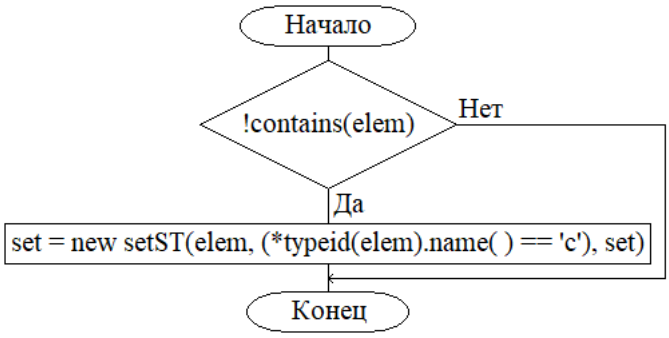
```
#include <iostream>
#include <typeinfo>
using std::cout;

class setOfCI
{
public:
    struct setST
    {
        int val;
        bool isChar;
        setST *next;
        setST() : val(0), next(NULL), isChar(false)
{}
        setST(int val) : val(val), next(NULL),
isChar(false) {}
        setST(int val, bool isChar) : val(val),
next(NULL), isChar(isChar) {}
        setST(int val, setST *next) : val(val),
next(next), isChar(false) {}
        setST(int val, bool isChar, setST *next) :
val(val), next(next), isChar(isChar) {}
    };
    setST *set = new setST();

    template <class intOrChar>
```

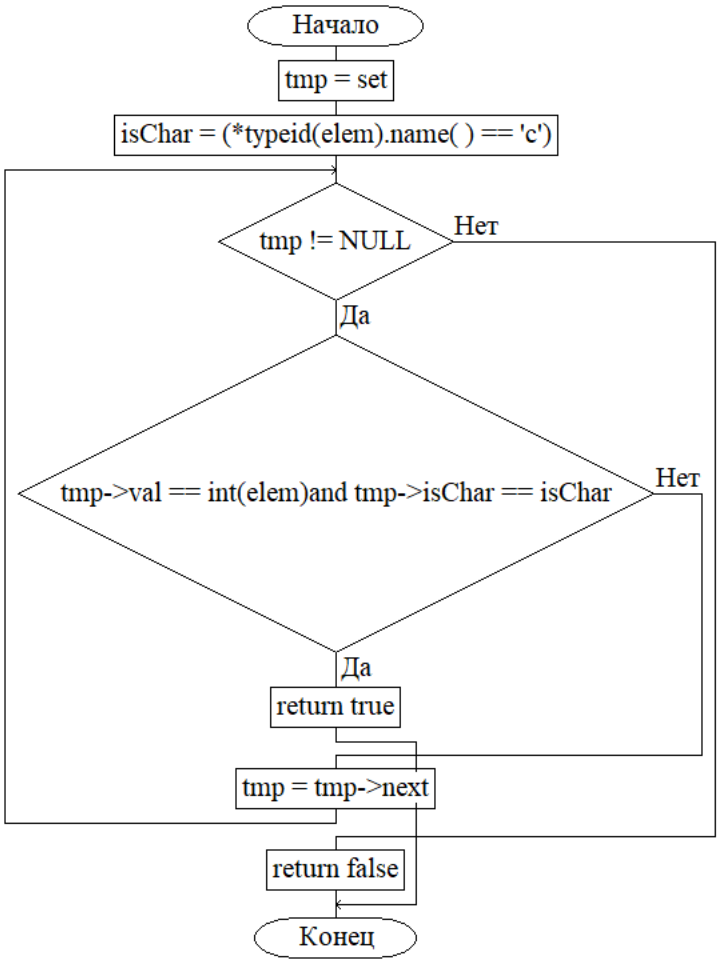
```
    bool contains(intOrChar elem)
    {
        setST *tmp = set;
        bool isChar = (*typeid(elem).name() == 'c');
        while (tmp != NULL)
        {
            if (tmp->val == int(elem) and tmp-
>isChar == isChar)
                return true;
            tmp = tmp->next;
        }
        return false;
    }

    template <class intOrChar>
    void add(intOrChar elem)
    {
        if (!contains(elem))
            set = new setST(elem,
(*typeid(elem).name() == 'c'), set);
    };

    template <class intOrChar>
    void del(intOrChar elem)
    {
        setST *tmp = set;
        setST *tmp2 = set;
        bool isChar = (*typeid(elem).name() == 'c');
        while (tmp != NULL)
        {
            if (tmp->val == int(elem) and tmp-
>isChar == isChar)
            {
```

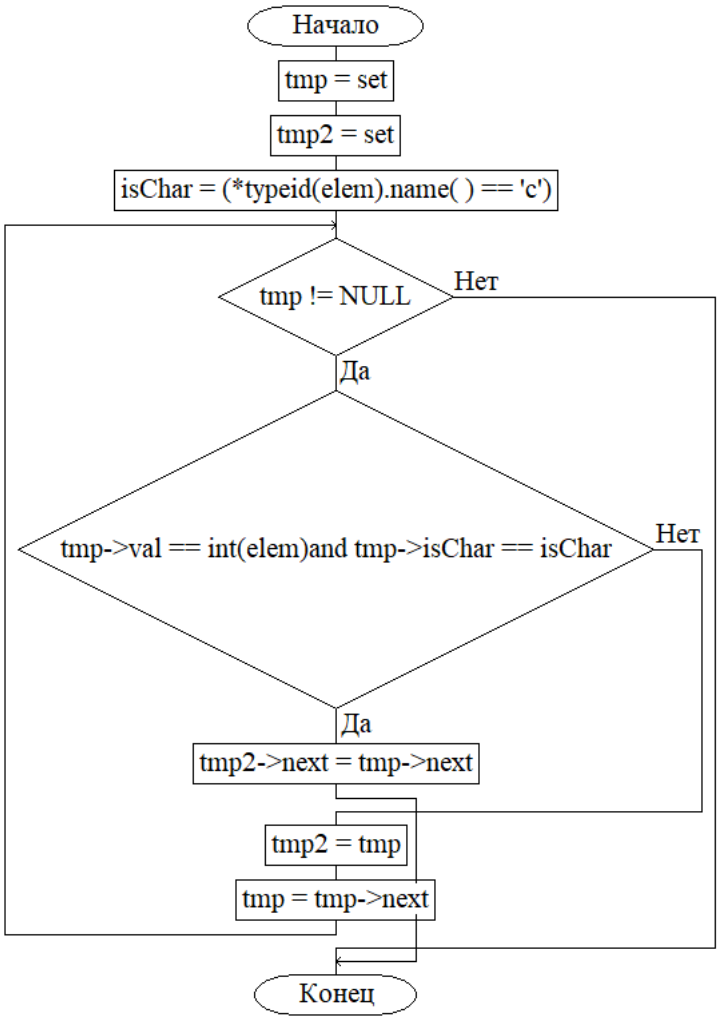
```
                tmp2->next = tmp->next;
                delete tmp;
                return;
            }
            tmp2 = tmp;
            tmp = tmp->next;
        }
        return;
    };

    void print()
    {
        setST *tmp = set;
        while (tmp != NULL)
        {
            if (tmp->next == NULL and tmp->val == 0)
                break;
            if (tmp->isChar)
                cout << char(tmp->val);
            else
                cout << tmp->val;
            cout << ' ';
            tmp = tmp->next;
        };
        cout << '\n';
    };
};

class setOfCIW : public setOfCI
{
public:
    int power()
    {
```

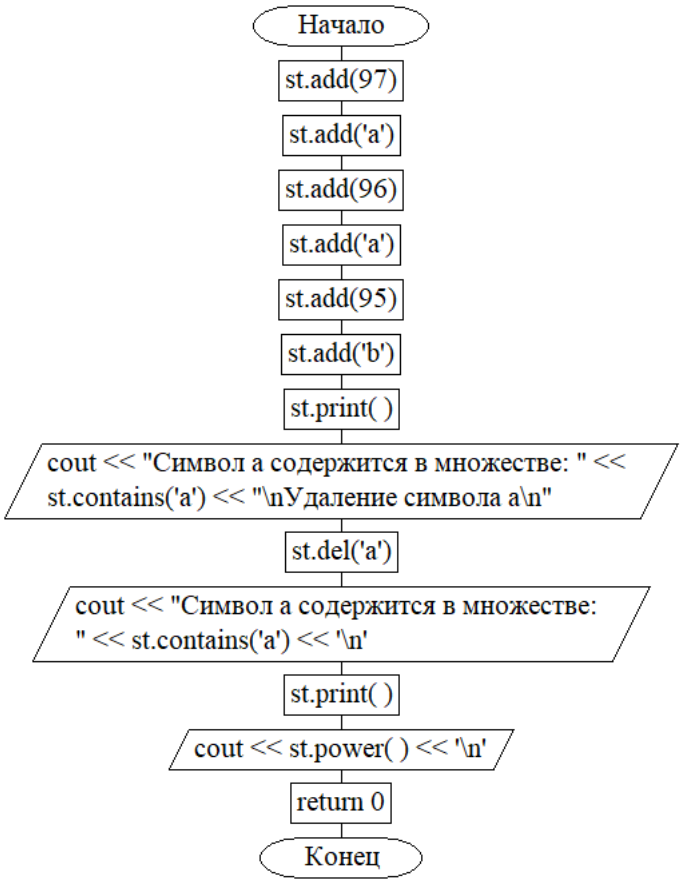
```
        setST *tmp = set;
        int i{};
        while (tmp != NULL)
        {
            i++;
            tmp = tmp->next;
        }
        return i;
    }
};

int main()
{
    setOfCIW st;
    st.add(97);
    st.add('a');
    st.add(96);
    st.add('a');
    st.add(95);
    st.add('b');
    st.print();
    cout << "Символ a содержится в множестве: " <<
st.contains('a')
         << "\nУдаление символа a\n";
    st.del('a');
    cout << "Символ a содержится в множестве: " <<
st.contains('a') << '\n';
    st.print();
    cout << st.power() << '\n';
    return 0;
}
```

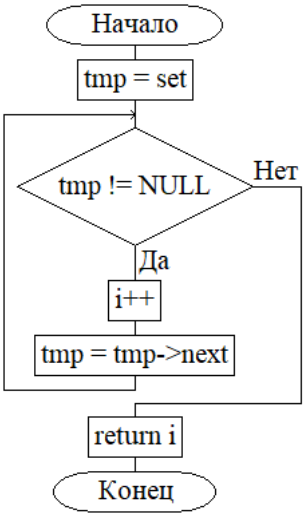

setOfCl
SetST
set
contains
add
print
setOfCIW
power


Начало
!contains(elem)
Нет
Да
set = new setST(elem, (*typeid(elem).name( ) == 'c'), set)
Конец


Начало
tmp = set
isChar = (*typeid(elem).name( ) == 'c')
tmp != NULL
Нет
Да
tmp->val == int(elem)and tmp->isChar == isChar
Нет
Да
return true
tmp = tmp->next
return false
Конец


Начало
tmp = set
tmp2 = set
isChar = (*typeid(elem).name( ) == 'c')
tmp != NULL
Нет
Да
tmp->val == int(elem)and tmp->isChar == isChar
Нет
Да
tmp2->next = tmp->next
tmp2 = tmp
tmp = tmp->next
Конец


Начало
st.add(97)
st.add('a')
st.add(96)
st.add('a')
st.add(95)
st.add('b')
st.print( )
cout << "Символ a содержится в множестве: " << st.contains('a') << "\nУдаление символа a\n"
st.del('a')
cout << "Символ a содержится в множестве: " << st.contains('a') << '\n'
st.print( )
cout << st.power( ) << '\n'
return 0
Конец


Начало
tmp = set
tmp != NULL
Нет
Да
i++
tmp = tmp->next
return i
Конец

```
b 95 96 a 97
Символ a содержится в множестве: 1
Удаление символа a
Символ a содержится в множестве: 0
b 95 96 97
5
```

Вывод: я научился организовывать собственный тип множество и использовать шаблоны C++.