*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)\_\_\_\_\_\_\_\_\_\_\_

**Отчет**

**по лабораторной работе № 8**

**Название лабораторной работы: Программирование с использованием классов в С++. 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 а содержится в множестве: " << st.contains('a')

<< "\nУдаление символа а\n";

st.del('a');

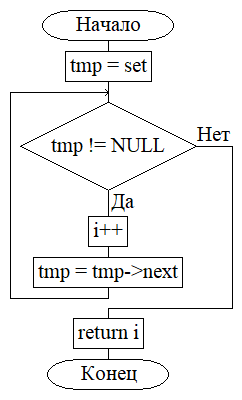
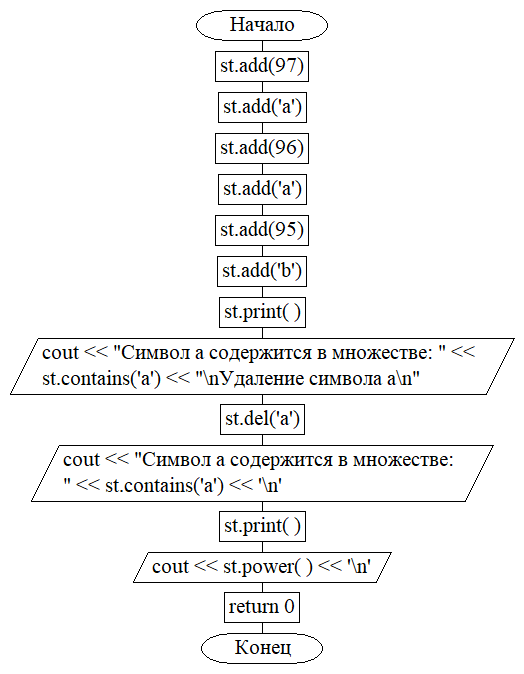
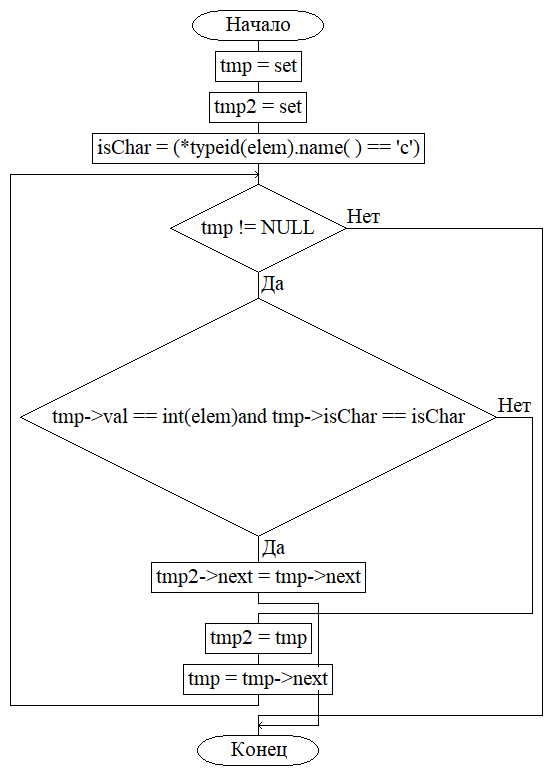
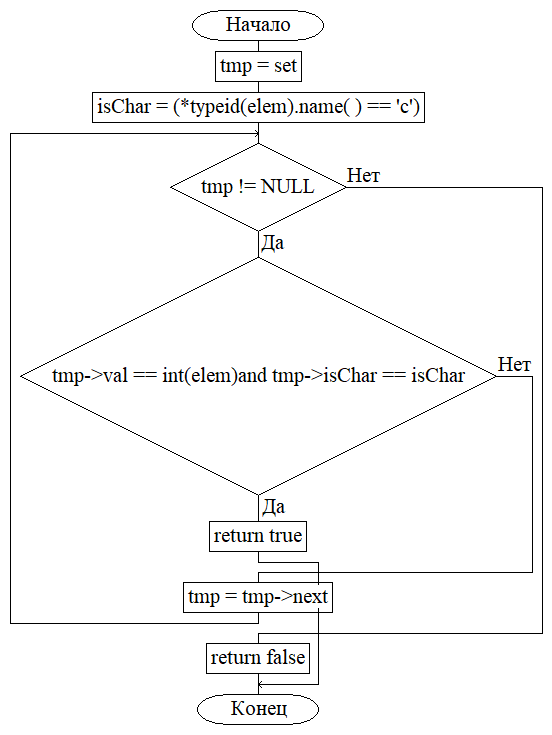
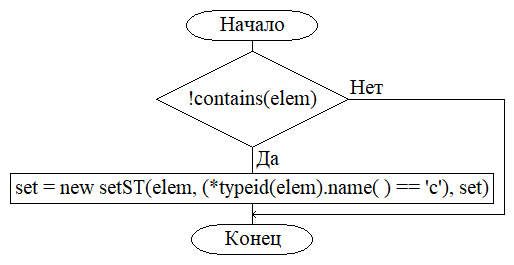
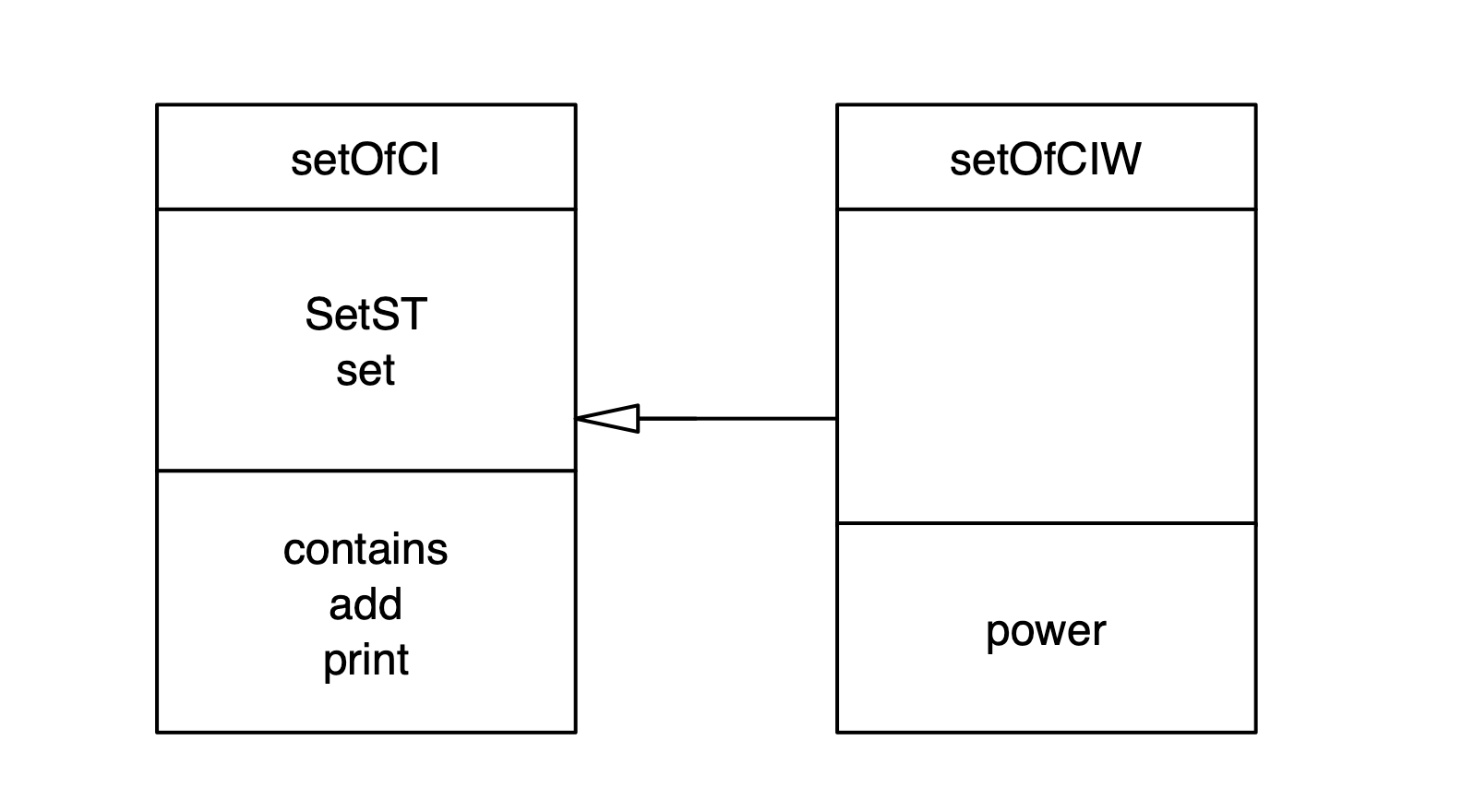
cout << "Символ а содержится в множестве: " << st.contains('a') << '\n';

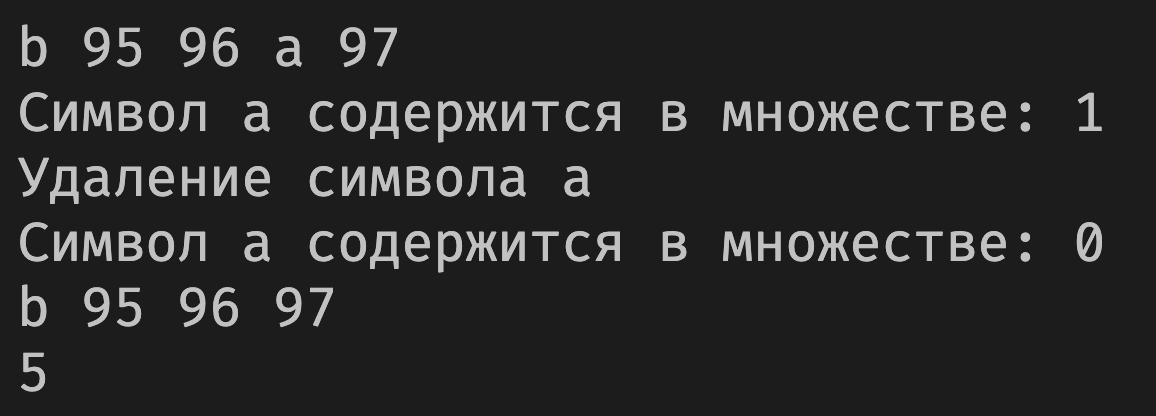
st.print();

cout << st.power() << '\n';

*return* 0;

}





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