



Министерство науки и высшего образования Российской Федерации
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 _____ «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 9
по курсу «Языки и методы программирования»
«Перегрузка операций»

Студент группы ИУ9-21Б Горбунов А. Д.

Преподаватель Посевин Д. П.

Москва 2023

1 Задание

`Set<Letter>` – множество «букв», представленных объектами некоторого класса `Letter`. (Требования к классу `Letter`: наличие операции «`==`», а также операции «`!`», некоторым образом вычисляющей так называемую «обратную букву». Для любой «буквы» `x` должно быть справедливо, что `!(! x) == x`.) Операции:

1. «`and=`» - добавление «букв» другого множества, сопровождаемое «редукцией», представляющей собой удаление из множества всех пар взаимнообратных «букв»;
2. «`and=`» – добавление в множество отдельной «буквы», также сопровождаемое «редукцией»;
3. «`!`» – замена всех «букв» в множестве на обратные им «буквы»;
4. «`==`», «`!=`». Класс `Set<Letter>` должен иметь конструктор без параметров, который создаёт пустое множество.

2 Результаты

Исходный код программы представлен в листинге 1, 2, 3

Листинг 1 — `main.cpp`

```
1 #include <iostream>
2 #include <set>
3 #include "Set.h"
4 using namespace std;
5 int main() {
6     Set<char> set1;
7     set1 &= 'a';
8     set1 &= 'b';
9     set1 &= 'c';
10    Set<char> set2;
11    set2 &= 'd';
12    set2 &= 's';
13    set2 &= 'c';
14    cout << "Set1: ";
15    for (auto letter : set1) {
16        cout << letter << " ";
17    }
18    cout << endl;
```

Листинг 2 — main.cpp(продолжение)

```
1      cout << "Set2: ";
2      for (auto letter : set2) {
3          cout << letter << " ";
4      }
5      cout << endl;
6      Set<char> set4;
7      set4 &= 'j';
8      set4 &= 'k';
9      set4 &= 'l';
10     cout << "Set4: ";
11     for (auto letter : set4) {
12         cout << letter << " ";
13     }
14     cout << endl;
15     (!set4);
16     cout << "!Set4: ";
17     for (auto letter : set4) {
18         cout << letter << " ";
19     }
20     cout << endl;
21     (!set4);
22     cout << "!(Set4): ";
23     for (auto letter : set4) {
24         cout << letter << " ";
25     }
26     cout << endl;
27     Set<char> set5;
28     set5 &= set1;
29     set5 &= set2;
30     cout << "Set5: ";
31     for (auto letter : set5) {
32         cout << letter << " ";
33     }
34     cout << std::endl;
35     cout << "Set1 == Set2: " << (set1 == set2) << endl;
36     cout << "Set1 != Set2: " << (set1 != set2) << endl;
37     return 0;
38 }
```

Листинг 3 — класс Set.h

```

1 #include <set>
2 #include <algorithm>
3 using namespace std;
4 template<typename T>
5 class Set {
6 private:
7     set<T> data;
8 public:
9     Set() {}
10    Set<T>& operator==(const Set<T>& other) {
11        for (const auto& item : other.data) {
12            this->data.insert(item);
13        }
14        for (auto it1 = this->data.begin(); it1 != this->data.end(); ++
it1) {
15            auto it2 = find_if(it1, this->data.end(), [&](const T& item)
{
16                return *it1 == !item;
17            });
18            if (it2 != this->data.end()) {
19                this->data.erase(it1);
20                this->data.erase(it2);
21                break;
22            }
23        }
24        return *this;
25    }
26    Set<T>& operator==(const T& item) {
27        this->data.insert(item);
28        auto it1 = this->data.find(item);
29        if (it1 != this->data.end()) {
30            auto it2 = find_if(it1, this->data.end(), [&](const T& item)
{
31                return *it1 == !item;
32            });
33            if (it2 != this->data.end()) {
34                this->data.erase(it1);
35                this->data.erase(it2);
36            }
37        }
38        return *this;
39    }
40    Set<T>& operator!() {
41        set<T> new_data;
42        for (const auto& item : this->data) {
43            new_data.insert(item ^ 15);
44        }
45        this->data = new_data;
46        return *this;
47    }
48    bool operator==(const Set<T>& other) const {
49        return this->data == other.data; }
50    bool operator!=(const Set<T>& other) const {
51        return this->data != other.data; }
52    typename set<T>::iterator begin() {
53        return data.begin(); }
54    typename set<T>::iterator end() {
55        return data.end(); }
56 };

```

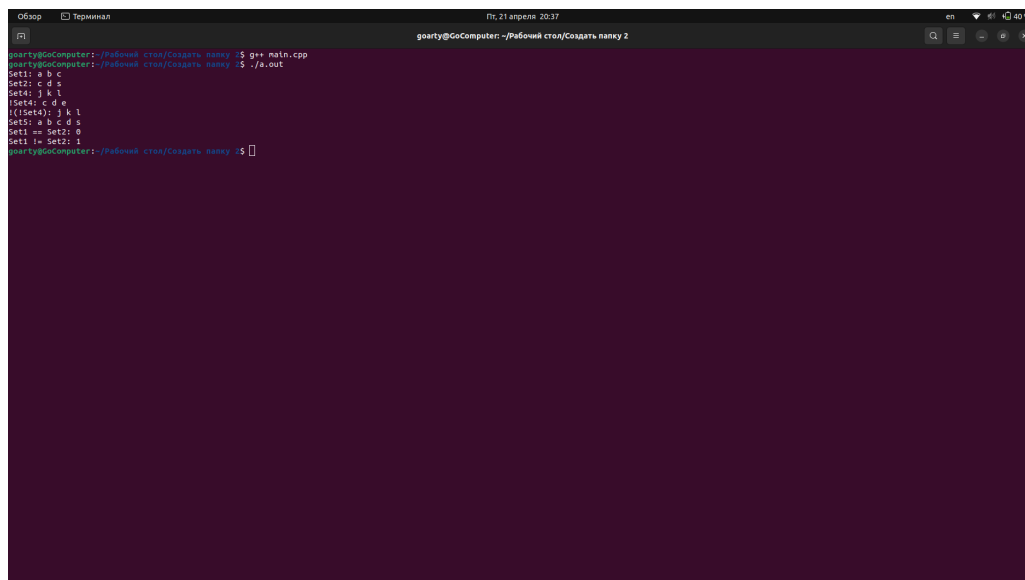
Результат запуска представлен на рисунке 1, 2, 3

```
main.cpp x Set.h x
1 #include <iostream>
2 #include <set>
3 #include "Set.h"
4 using namespace std;
5 int main() {
6     Set<char> set1;
7     set1.insert('a');
8     set1.insert('b');
9     set1.insert('c');
10
11     Set<char> set2;
12     set2.insert('d');
13     set2.insert('e');
14     set2.insert('f');
15
16     cout << "Set1: ";
17     for (auto letter : set1) {
18         cout << letter << " ";
19     }
20     cout << endl;
21
22     cout << "Set2: ";
23     for (auto letter : set2) {
24         cout << letter << " ";
25     }
26     cout << endl;
27
28     Set<char> set4;
29     set4.insert('j');
30     set4.insert('k');
31     set4.insert('l');
32
33     cout << "Set4: ";
34     for (auto letter : set4) {
35         cout << letter << " ";
36     }
37     cout << endl;
38     (*set4);
39
40     cout << "Set4: ";
41     for (auto letter : set4) {
42         cout << letter << " ";
43     }
44     cout << endl;
45     (*set4);
46     cout << "!(Set4): ";
47     for (auto letter : set4) {
48         cout << letter << " ";
49     }
50     cout << endl;
51
52     Set<char> set5;
53     set5.insert('a');
54     set5.insert('b');
55
56     cout << "Set5: ";
57     for (auto letter : set5) {
58         cout << letter << " ";
59     }
60     cout << endl;
61
62     cout << "Set1 == Set2: " << (set1 == set2) << endl;
63     cout << "Set1 != Set2: " << (set1 != set2) << endl;
64     return 0;
65 }
```

Рис. 1 — Реализация main.cpp

```
main.cpp x Set.h x
1 #include <iostream>
2 #include <algorithm>
3 using namespace std;
4 template<typename T>
5 class Set {
6 private:
7     Set<T> data;
8 public:
9     Set() {}
10
11     Set<T> operator=(const Set<T> &other) {
12         for (const auto item : other.data) {
13             this->data.insert(item);
14         }
15         for (auto it1 = this->data.begin(); it1 != this->data.end(); ++it1) {
16             auto it2 = find_if(it1, this->data.end(), [&](const T& item) {
17                 return it1 == item;
18             });
19             if (it2 != this->data.end()) {
20                 this->data.erase(it1);
21                 this->data.erase(it2);
22                 break;
23             }
24         }
25     }
26
27     return *this;
28 }
29
30 Set<T> operator=(const T& item) {
31     this->data.insert(item);
32
33     auto it1 = this->data.find(item);
34     if (it1 != this->data.end()) {
35         auto it2 = find_if(it1, this->data.end(), [&](const T& item) {
36             return it1 == item;
37         });
38         if (it2 != this->data.end()) {
39             this->data.erase(it1);
40             this->data.erase(it2);
41         }
42     }
43     return *this;
44 }
45
46 Set<T> operator() {
47     Set<T> new_data;
48     for (const auto item : this->data) {
49         new_data.insert(item * 10);
50     }
51     this->data = new_data;
52     return *this;
53 }
54
55 bool operator==(const Set<T> &other) const {
56     return this->data == other.data;
57 }
58 bool operator!=(const Set<T> &other) const {
59     return this->data != other.data;
60 }
61
62 typename Set<T>::iterator begin() {
63     return data.begin();
64 }
65
66 typename Set<T>::iterator end() {
67     return data.end();
68 }
69 };
70
71 }
```

Рис. 2 — Реализация и Set.h



```
Обзор Терминал Пн, 21 апреля 2023 г. 61% 40%
goarty@GoComputer: ~/Рабочий стол/Создать папку 2
goarty@GoComputer: ~/Рабочий стол/Создать папку 2$ g++ main.cpp
goarty@GoComputer: ~/Рабочий стол/Создать папку 2$ ./a.out
Set1: a b c
Set2: c d s
Set4: j k l
Set4: c d e
Set4: j k l
Set5: a b c d s
Set1 = Set2: d
Set1 != Set2: l
```

Рис. 3 — Работа программы