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

Лабораторная работа №5
по дисциплине: **ООТПиСТ**
Тема: СТАНДАРТНАЯ БИБЛИОТЕКА ШАБЛОНОВ

Выполнил
студент 2 курса
Корнасевич И. Д.

Проверил
Миндер А. В.

Брест 2021

Задание1

1. Создать контейнер в соответствии с вариантом задания и заполнить его данными типа int
2. Удалить элементы из контейнера по предикату
3. Создать второй контейнер
4. Изменить первый контейнер, удалив из него n элементов после заданного и добавив затем в него все элементы из второго контейнера
5. Добавить все элементы из второго контейнера в первый

Задание2

1. Создать контейнер в соответствии с вариантом задания и заполнить его данными типа int
2. Отсортировать контейнер по убыванию
3. Создать второй контейнер
4. Скопировать элементы, удовлетворяющие предикату, из первого контейнера во второй
5. Отсортировать оба контейнера
6. Получить третий контейнер слиянием первых двух
7. Подсчитать, сколько элементов, удовлетворяющих предикату, содержит третий контейнер

main.cpp

```cpp
#include <iostream>
#include <vector>
#include <algorithm>
#include <list>
#include <numeric>

void printVector(std::vector<int> &v, const std::string& message) {
    std::cout << message << " ";
    for (int i : v) {
        std::cout << i << ", ";
    }
    std::cout << std::endl;
}

void task1() {
    std::vector<int> v(20);
    std::iota(v.begin(), v.end(), 0);
    printVector(v, "v:");

```

```
20      auto b = std::remove_if(v.begin(), v.end(), [](int i) { return i
    > 10; });
21      v.erase(b, v.end());
22
23      printVector(v, "v:");
24
25      std::list<int> list(20);
26      std::iota(list.begin(), list.end(), 50);
27
28      v.erase(v.begin() + 5, v.end());
29      printVector(v, "v after erasing:");
30      for (int i : list) {
31          v.push_back(i);
32      }
33
34      printVector(v, "v after merging:");
35  }
36
37  void task2() {
38      std::vector<int> v1(20);
39      std::iota(v1.begin(), v1.end(), 0);
40      printVector(v1, "v1:");
41      std::sort(v1.begin(), v1.end(), [](int a, int b) { return a >= b;
        });
42      printVector(v1, "v1 after reverse sort:");
43      std::vector<int> v2(0);
44      std::copy_if(v1.begin(), v1.end(), std::back_inserter(v2), [](int
        a) { return a >= 10; });
45      printVector(v2, "v2:");
46      std::sort(v1.begin(), v1.end());
47      std::sort(v2.begin(), v2.end());
48      printVector(v1, "v1 after sort:");
49      printVector(v2, "v2 after sort:");
50      std::vector<int> v3(0);
51      std::set_union(v1.begin(), v1.end(), v2.begin(), v2.end(), std::
    back_inserter(v3));
52      printVector(v3, "v3 after union:");
53      int count = std::count_if(v3.begin(), v3.end(), [](int x){return
    x < 4;});
54      std::cout << count << " elements => x < 4:";
55  }
56
57  int main() {
58      std::cout << "\ntask 1\n\n";
59      task1();
60      std::cout << "\ntask 2\n\n";
61      task2();
62  }
```

results.txt

```
1  task 1
2
3  v: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
       19,
4  v: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
```

```
v after erasing: 0, 1, 2, 3, 4,
v after merging: 0, 1, 2, 3, 4, 50, 51, 52, 53, 54, 55, 56, 57, 58,
    59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69,

task 2

v1: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
    19,
v1 after reverse sort: 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8,
   7, 6, 5, 4, 3, 2, 1, 0,
v2: 19, 18, 17, 16, 15, 14, 13, 12, 11, 10,
v1 after sort: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
   16, 17, 18, 19,
v2 after sort: 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
v3 after union: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
    16, 17, 18, 19,
4 elements => x < 4:
```

Вывод: Я работе я использовал generic контейнеры std::vector и std::list. Я изучил функции std::iota для заполнения контейнера; std::remove, std::remove_if и std::erase для удаления элементов; std::sort для сортировки по предикату и без него; std::copy_if для копирования по предикату; std::set_union для слияния двух контейнеров.