Министерство образование Республики Беларусь

Учреждение образования «Полоцкий государственный университет»

Информационно-технологический факультет

Кафедра технологий программирования

Лабораторная работа №6 по теме «СТАНДАРТНАЯ БИБЛИОТЕКА ШАБЛОНОВ»

По дисциплине «Объектно-ориентированные технологии программирования и стандарты проектирования»

Выполнил: Ролевич М.А.

Группа 19-ИТ-1.

Проверил: Васильева Д. М.

Новополоцк

2021 г.

**Вариант 3**

**Цель работы:** получить практические навыки использования стандартной библиотеки шаблонов при написании кода на языке C++.

**Основное содержание работы.**

Написать интерактивную программу, выполняющую команды, вводимые пользователем с клавиатуры.

**Порядок выполнения работы.**

Написать и отладить три программы. Первая программа

демонстрирует использование контейнерных классов для хранения

встроенных типов данных. Вторая программа демонстрирует

использование контейнерных классов для хранения пользовательских типов

данных. Третья программа демонстрирует использование алгоритмов STL.

**В программе № 1** выполнить следующее:

1. Создать объект-контейнер в соответствии с вариантом задания и заполнить

его данными, тип которых определяется вариантом задания.

2. Просмотреть контейнер.

3. Изменить контейнер, удалив из него одни элементы и изменив другие.

4. Просмотреть контейнер, используя для доступа к его элементам

итераторы.

**В программе № 2** то же самое, но для данных пользовательского типа.

**В программе № 3** выполнить следующее:

1. Создать контейнер, содержащий объекты пользовательского типа. Тип

контейнера выбирается в соответствии с вариантом задания.

2. Отсортировать его по убыванию элементов.

3. Просмотреть контейнер.

4. Используя подходящий алгоритм, найти в контейнере элемент,

удовлетворяющий заданному условию.

5. Переместить элементы, удовлетворяющие заданному условию в другой

(предварительно пустой) контейнер.

6. Просмотреть второй контейнер.

7. Отсортировать первый и второй контейнеры по возрастанию элементов.

8. Просмотреть их.

В моём варианте предложены значения:



**Задание 1.** Программа №1:

1. void OneQuestion()
2. {
3. deque<int> count;
4. deque<int> ::iterator it;
5. for (int i = 0; i < 10; i++)
6. {
7. if (i < 3)
8. {
9. count.push\_back(111);
10. continue;
11. }
12. if (i < 6)
13. {
14. count.push\_back(222);
15. continue;
16. }
17. if (i < 9)
18. {
19. count.push\_back(333);
20. continue;
21. }
22. count.push\_back(444);
23. }
24. for (int i = 0; i < count.size(); i++)
25. {
26. cout << count[i] << " ";
27. }
28. cout << endl;
29. replace(count.begin(), count.end(), 111, 222);
30. for (int i = 0; i < count.size(); i++)
31. {
32. if (count[i] == 333)
33. {
34. count.erase(count.begin() + i);
35. i--;
36. }
37. if (count[i] == 444)
38. {
39. count.erase(count.begin() + i);
40. i--;
41. }
42. }
43. it = count.begin();
44. while (it != count.end())
45. {
46. cout << \*it << " ";
47. it++;
48. }
49. }

**Задание 2.** Программа №2:

1. void TwoQuestion()
2. {
3. deque<Car> count;
4. deque<Car> ::iterator it;
5. for (int i = 0; i < 10; i++)
6. {
7. if (i < 5)
8. {
9. Car d;
10. count.push\_back(d);
11. continue;
12. }
13. Car c(500, 101);
14. count.push\_back(c);
15. }
16. for (int i = 0; i < count.size(); i++)
17. {
18. count[i].print();
19. cout << endl;
20. }
21. cout << endl;
22. cout << "OMG 4TO ETO TAKOE" << endl;
23. //replace(count.begin(), count.end(), "abc", "xyz");
24. for (int i = 0; i < count.size(); i++)
25. {
26. if (count[i].getNumber() == 0)
27. {
28. count.erase(count.begin() + i);
29. i--;
30. }
31. }
32. for (int i = 0; i < count.size(); i++)
33. {
34. if (count[i].getNumber() == 500)
35. {
36. count[i].setNumber2(96);
37. count[i].setNumber(69);
38. }
39. }
40. it = count.begin();
41. while (it != count.end())
42. {
43. (\*it).print();
44. it++;
45. cout << endl;
46. }
47. }

**Задание 3.** Программа №3:

1. void ThreeQuestion()
2. {
3. deque<Car> count;
4. for (int i = 0; i < 3; i++)
5. {
6. Car d(i \* 100, 1);
7. count.push\_back(d);
8. }
9. for (int i = 0; i < 5; i++)
10. {
11. Car d(i \* 50, 2);
12. count.push\_back(d);
13. }
14. for (int i = 0; i < 4; i++)
15. {
16. Car d(i \* 1000, 3);
17. count.push\_back(d);
18. }
19. for (int i = 0; i < count.size(); i++)
20. {
21. count[i].print();
22. cout << endl;
23. }
24. cout << "------------------------" << endl;
25. sort(count.rbegin(), count.rend());
26. for (int i = 0; i < count.size(); i++)
27. {
28. count[i].print();
29. cout << endl;
30. }
31. cout << "------------------------" << endl;
32. deque<Car> ncount;
33. for (int i = 0; i < count.size(); i++)
34. {
35. if (count[i].getNumber() < 500)
36. {
37. ncount.push\_back(count[i]);
38. count.erase(count.begin() + i);
39. i--;
40. }
41. }
42. sort(count.begin(), count.end());
43. for (int i = 0; i < count.size(); i++)
44. {
45. count[i].print();
46. cout << endl;
47. }
48. cout << "-----------------------" << endl;
49. sort(ncount.begin(), ncount.end());
50. for (int i = 0; i < ncount.size(); i++)
51. {
52. ncount[i].print();
53. cout << endl;
54. }
55. }

**Задание 4.** Пользовательский тип данных в заданном конструкторе для 2,3 программы:

1. class Car
2. {
3. public:
4. Car();
5. Car(int, int);
6. ~Car();
7. void print();
8. int getNumber();
9. int getNumber2();
10. void setNumber(int N);
11. void setNumber2(int S);
12. bool operator<(const Car&);
13. bool operator==(const Car&);
14. private:
15. int number;
16. int name;
17. };
18. Car::Car()
19. {
20. number = 0;
21. name = 0;
22. }
23. Car::Car(int N, int S)
24. {
25. number = N;
26. name = S;
27. }
28. Car::~Car()
29. {
30. }
31. void Car::print()
32. {
33. cout << name << "========" << number;
34. }
35. int Car::getNumber()
36. {
37. return number;
38. }
39. int Car::getNumber2()
40. {
41. return name;
42. }
43. void Car::setNumber(int N)
44. {
45. number = N;
46. }
47. void Car::setNumber2(int S)
48. {
49. name = S;
50. }
51. bool Car::operator<(const Car& d)
52. {
53. return this->number < d.number;
54. }
55. bool Car::operator==(const Car& d)
56. {
57. return this->number == d.number && this->name == d.name;
58. }