Министерство науки и высшего образования Российской Федерации  
Пензенский государственный университет  
Кафедра вычислительная техника

**ОТЧËТ**  
по лабораторной работе №6  
по курсу «Объектно-ориентированное программирование»  
на тему «Стандартная библиотека шаблонов (STL) в языке C++»

Выполнили студенты группы 22ВВП1:  
Ипполитов И. Д.

Хоссейни Нежад С. А. С. М.

Приняли:  
Евсеева Ю.И.  
Гудков А. А.

Пенза 2024

**Название**

Стандартная библиотека шаблонов (STL) в языке C++

**Цель работы**

Изучить стандартную библиотеку шаблонов (STL) в языке C++.

**Лабораторное задание**

Используйте шаблон map для исключения повторов среди множества целых чисел.

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 заменив другие.

4. Просмотреть контейнер, используя для доступа к его элементам итераторы.

5. Создать второй контейнер этого же класса и заполнить его данными того же типа, что и первый

контейнер.

6. Изменить первый контейнер, удалив из него n элементов после заданного и добавив затем в

него все элементы из второго контейнера.

7. Просмотреть первый и второй контейнеры.

**Листинг**

#include <iostream>

#include <windows.h>

#include <map>

using namespace std;

void printMap(map<int, int>& mp)

{

map <int, int> ::iterator it = mp.begin();

cout << "map:" << endl;

for (int i = 0; it != mp.end(); it++, i++)

{

cout << i << ") Ключ: " << it->first << ", значение: " << it->second << endl << endl;

}

}

void fillMap(map<int, int>& mp)

{

cout << "Введите количество элементов: ";

int n;

cin >> n;

for (int i = 0; i < n; i++)

{

int k, m;

cout << "Введите ключ для " << i << " элемента: ";

cin >> k;

cout << "Введите значение " << i << " элемента: ";

cin >> m;

mp.insert(make\_pair(k, m));

}

cout << endl;

}

void eraseElementMap(map<int, int>& mp)

{

int n, key;

map <int, int> ::iterator it;

cout << "Введите количество удаляемых элементов: ";

cin >> n;

for (int i = 0; i < n; i++)

{

cout << "Введите ключ удаляемого элемента: ";

cin >> key;

it = mp.find(key);

if (it != mp.end())

mp.erase(it);

else

cout << "Вы ввели неправильный ключ!";

}

}

void replacingАnЕlementMap(map<int, int>& mp)

{

int n, key, meaning;

map <int, int> ::iterator it;

cout << "Введите количество заменяемых элементов: ";

cin >> n;

for (int i = 0; i < n; i++)

{

cout << "Введите ключ заменяемого элемента: ";

cin >> key;

it = mp.find(key);

if (it != mp.end())

{

mp.erase(it);

cout << "Введите значение заменяемого элемента: ";

cin >> meaning;

mp.insert(make\_pair(key, meaning));

}

else

cout << "Вы ввели неправильный ключ!";

}

}

void deletingItemsAfterTheKey(map<int, int>& mp)

{

int n, key;

map <int, int> ::iterator it;

map <int, int> ::iterator it2;

cout << "Введите после какого ключа удалять элементы: ";

cin >> key;

cout << "Введите сколько элементов удалить: ";

cin >> n;

cout << endl;

it = mp.begin();

int j = 0;

while (it != mp.end())

{

if (it->first > key && j < n)

{

it2 = it;

it++;

mp.erase(it2);

j++;

continue;

}

it++;

}

}

void addMap2ToMap1(map<int, int>& mp2, map<int, int>& mp)

{

map <int, int> ::iterator it;

it = mp2.begin();

for (; it != mp2.end(); it++)

{

mp.insert(make\_pair(it->first, it->second));

}

}

int main()

{

setlocale(LC\_ALL, "rus");

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

map <int, int> mp;

map <int, int> mp2;

cout << "Заполнение 1 контейнера" << endl;

fillMap(mp);

printMap(mp);

cout << "Изменение 1 контейнера" << endl;

eraseElementMap(mp);

replacingАnЕlementMap(mp);

printMap(mp);

cout << "Заполнение 2 контейнера" << endl;

fillMap(mp2);

cout << "Изменение 1 контейнера" << endl;

deletingItemsAfterTheKey(mp);

addMap2ToMap1(mp2, mp);

cout << "Map1:" << endl;

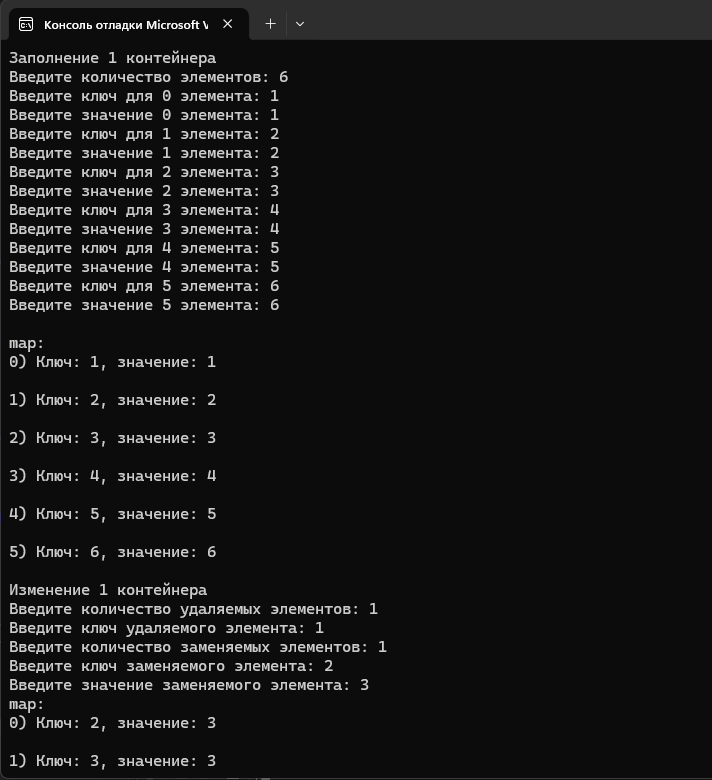
printMap(mp);

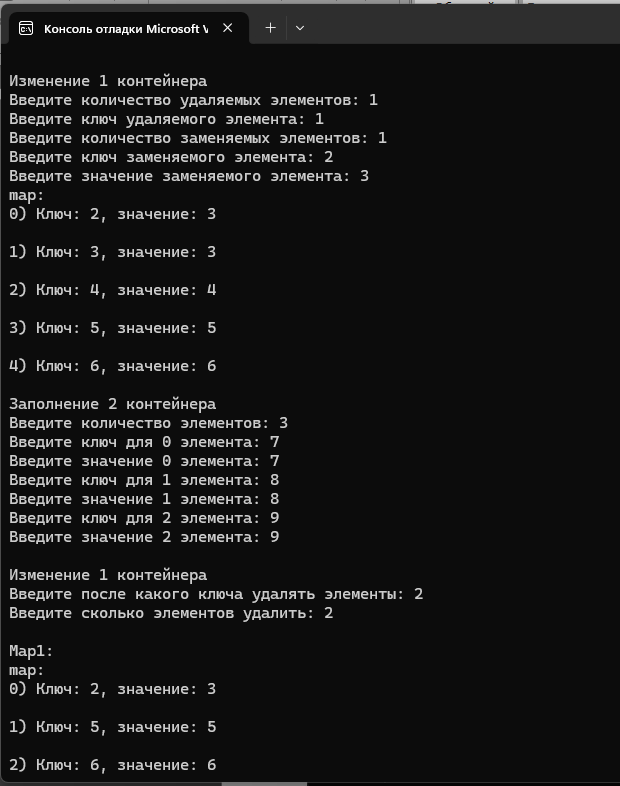
cout << "Map2:" << endl;

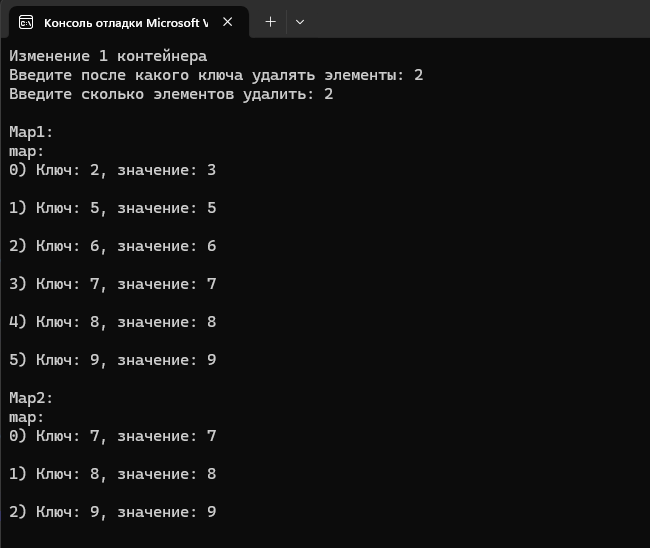
printMap(mp2);

}

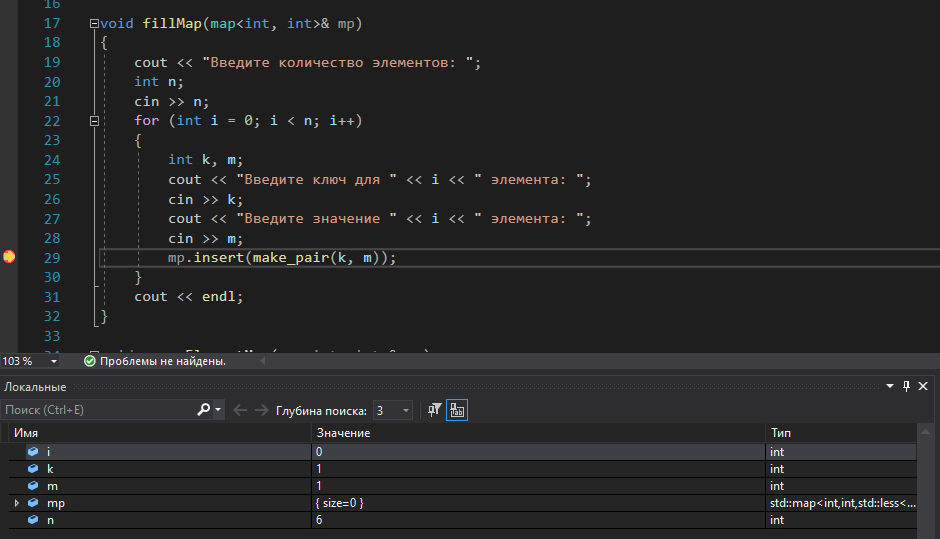
**Результат работы программы**

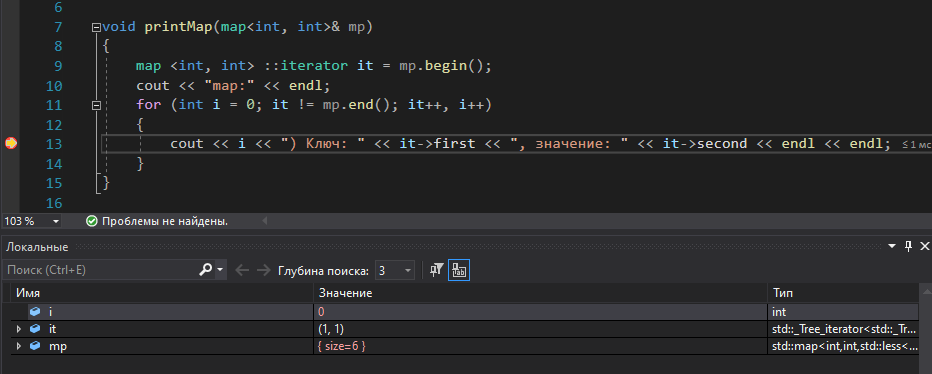


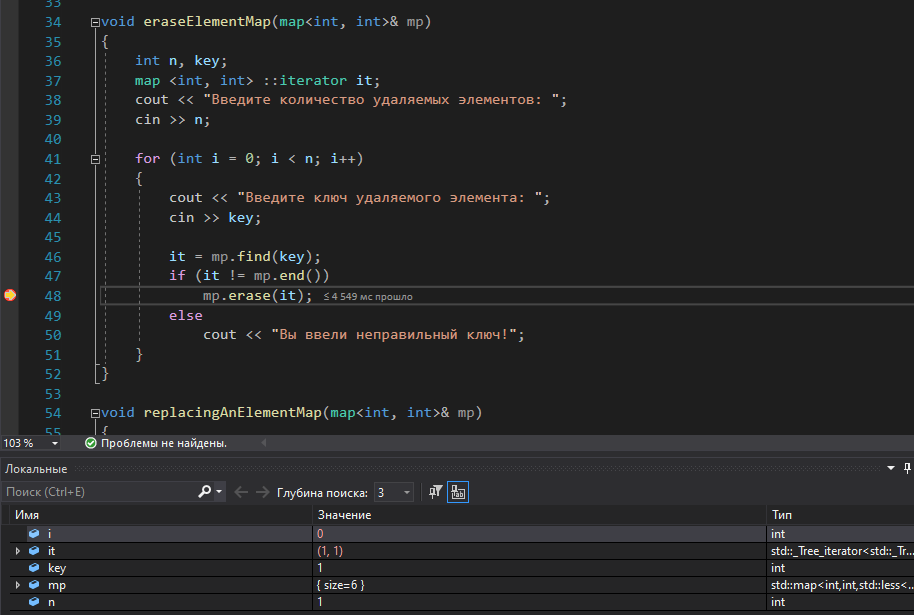


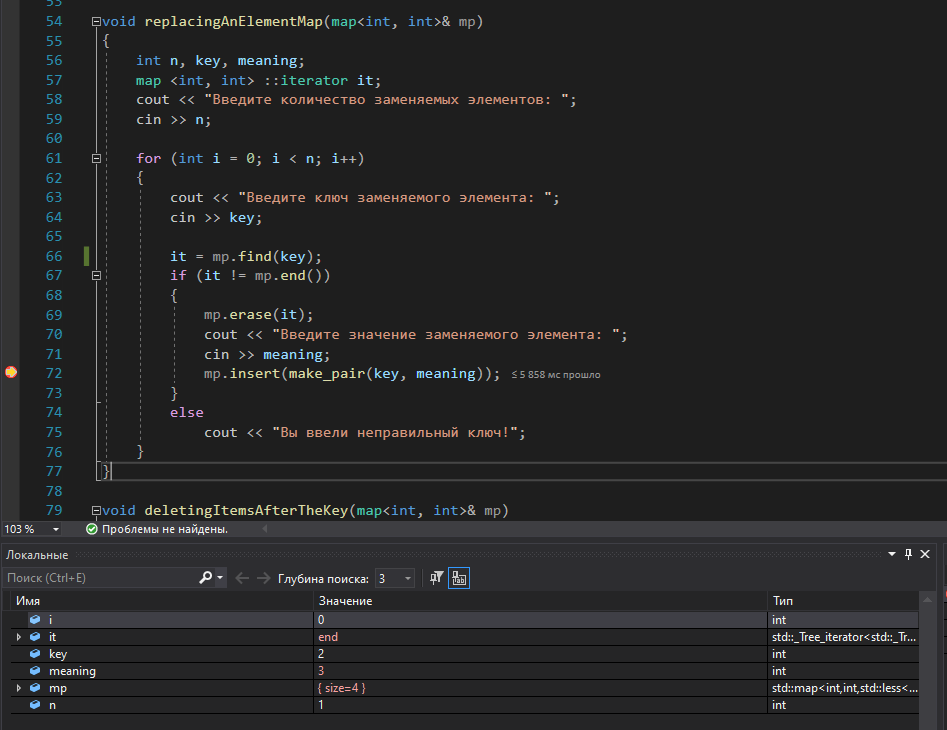


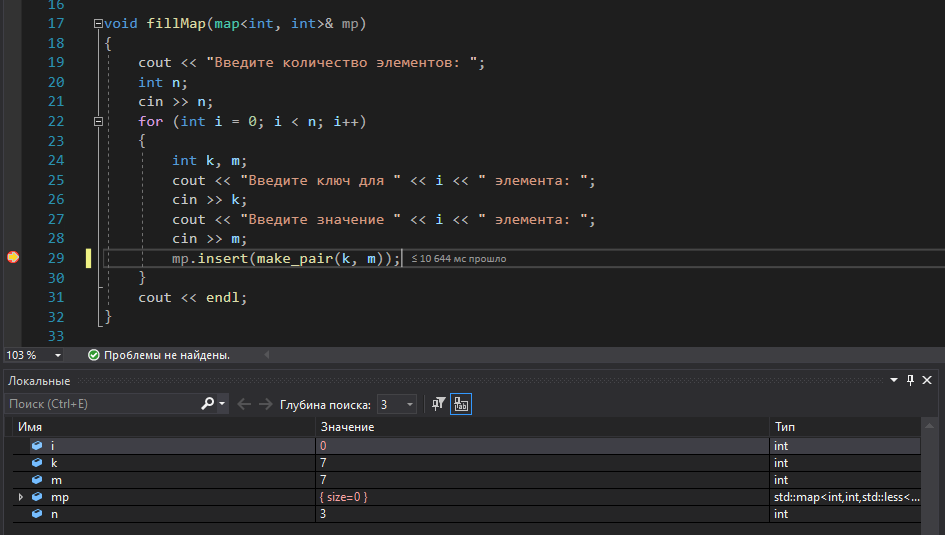
**Протокол трассировки программы**

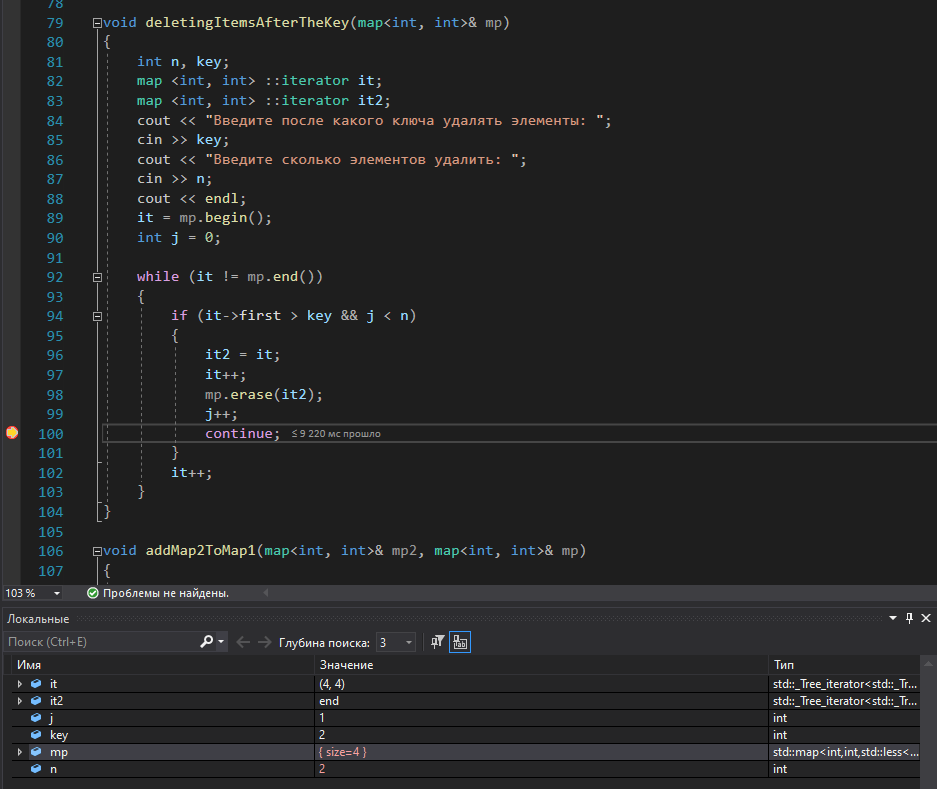


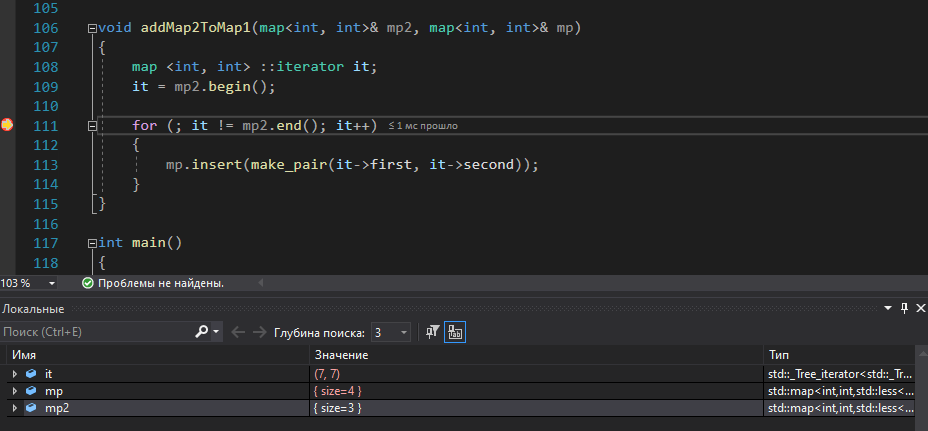












**Вывод**

Мы изучили стандартную библиотеку шаблонов (STL) в языке C++.