*Федеральное государственное бюджетное образовательное учреждениевысшего профессионального образования*

|  |  |
| --- | --- |
|  | ***«Московский государственный технический университет***  ***имени Н.Э. Баумана»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ: Информатики и систем управления

КАФЕДРА: Компьютерные системы и сети

**Отчет**

**по усложнённой лабораторной работе №** 7

**Название лабораторной работы:** *Программирование с использованием контейнеров STL и Qt*

**Дисциплина:** Объектно-ориентированное программирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент гр. ИУ6-22 |  |  | | Боярских Никита |
|  | (Подпись, дата) | |  | (И. О. Фамилия) |
| Преподаватель |  | |  | Черноусова Т. Г. |
|  | (Подпись, дата) | |  | (И. О. Фамилия) |

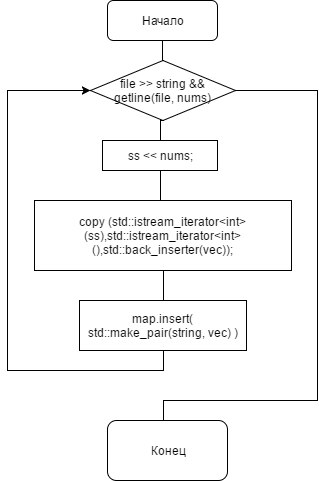
**Вариант 1**

Москва, 2017

**Задание:**

Разработать программу построения индекса записей в файле с использованием stl::map и QMap. Минимально запись состоит из строки-имени, массива из 100 целых числе. Реализовать свои функции сравнения, игнорирующие различия в регистре символов. Сравнить скорость работы разных реализаций в отладочном и финальном режиме.

Для поставленной задачи были разработан алгоритм, представленный на схеме (см. рис. 1)



*Рисунок 1 - схема алгоритма программы построения индекса записей в файле*

Код программы построения индекса записей в файле (версия с QMap)

#include <QCoreApplication>

#include <QMap>

#include <QFile>

#include <QString>

#include <QTextStream>

#include <QStringList>

#include <QVector>

class MyString : public QString

{

public:

bool operator <(const MyString &str) const;

};

(Продолжение кода на следующей странице)

(Продолжение кода)

bool MyString::operator <(const MyString &str) const

{

return this->toLower() < str.toLower();

}

int main()

{

QTextStream cout(stdout);

QFile file("data.tsv");

if( !file.open(QFile::ReadOnly | QIODevice::Text) ) {

cout <<"Error while opening file!"<< endl;

}

QTextStream ifs(&file);

QMap< MyString, QVector<int>> map;

while( !ifs.atEnd() ) {

QVector<int> vec;

MyString str;

ifs >> str;

QStringList list = ifs.readLine().split('\t');

QString i;

foreach (i, list) {

vec.append( i.toInt() );

}

map.insert(str, vec);

}

return 0;

}

Код программы построения индекса записей в файле (версия с std::map)

#include <iostream>

#include <map>

#include <fstream>

#include <string>

#include <vector>

#include <sstream>

#include <algorithm>

#include <iterator>

#include <cctype>

struct comparator {

bool operator() ( std::string const &a, std::string const &b ) const

{

std::string new\_a(a);

std::string new\_b(b);

std::transform( new\_a.begin(), new\_a.end(), new\_a.begin(), tolower );

std::transform( new\_b.begin(), new\_b.end(), new\_b.begin(), tolower );

return new\_a < new\_b;

}

};

int main( int argc, char \*argv[] )

{

std::ifstream file("data.tsv");

std::string string, nums;

std::map< std::string, std::vector<int>, comparator > map;

(Продолжение кода на следующей странице)

(Продолжение кода)

while( file >> string && getline(file, nums) ) {

std::vector<int> vec;

std::stringstream ss;

ss << nums;

std::copy

(

std::istream\_iterator<int>(ss),

std::istream\_iterator<int>(),

std::back\_inserter(vec)

);

map.insert( std::make\_pair(string, vec) );

}

return 0;

}

Проведено тестирование программ и сравнение их времени работы (см. табл. 1)

Таблица 1 - сравнение времени работы программы в разных режимах и в разных версиях (в секундах)

|  |  |  |
| --- | --- | --- |
|  | Отладочный режим | Финальный режим |
| std::map | 0,07 | 0,05 |
| QMap | 0,15 | 0,12 |

**Вывод:**

1) Разработаналгоритм для решения поставленной задачи. Составлена егосхема в среде [draw.io](http://draw.io)(рис. 1)

2) Создан код программы в среде Qt Creator

3) Проведено сравнение времени работы программы при использовании контейнеров std::map и QMap (см. табл.1)