**МИНОБРНАУКИ РОССИИ**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»  
(БГТУ им. В.Г.Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №1

Дисциплина: Обьектно-ориентированное программирование

по теме «Знакомство с интегрированной средой разработки (ИСР) Microsoft Visual Studio»

Выполнил: ст. группы ВТ-22  
Ковалев Павел ВТ-22

Проверил: Буханов Д.Г.

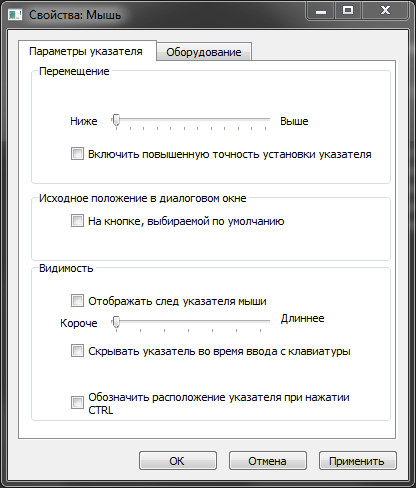
Белгород 2019

Вариант 3

а. B текстовом файле хранятся квадратные вещественные матрицы порядка n (n - const) по строкам. Преобразовать файл, удалив из каждой матрицы первую строку и первый столбец.

б. Дан файл целых чисел. Нулевых компонент в файле нет. Число отрицательных компонент равно числу положительных. Преобразовать файл таким образом, чтобы сначала были положительные числа, а затем отрицательные. Порядок следования как положительных, так и отрицательных чисел сохранить.

Форма «Свойства: Мышь», вкладки «Параметры указателя» и «Оборудование» (панель управления)



Task 1

#include <QCoreApplication>

#include <QTextCodec>

#include <iostream>

using namespace std;

#include <fstream>

/\*а. B текстовом файле хранятся квадратные вещественные матрицы порядка n (n - const) по строкам.

\* Преобразовать файл, удалив из каждой матрицы первую строку и первый столбец.\*/

//cout << QString::fromUtf8("").toLocal8Bit().data();

void **creat\_file**(){

ofstream file;

file.open("test.txt");

unsigned order;

unsigned count\_matr;

cout << QString::fromUtf8("Введите порядок матрицы:").toLocal8Bit().data();

cin >> order;

cout << QString::fromUtf8("Введите количество матриц:").toLocal8Bit().data();

cin >> count\_matr;

double number;

file << order\*order << endl;

for (unsigned i=0;i<count\_matr;i++) {

for (unsigned j = 0; j<order\*order;j++) {

cin >> number;

file << number << " ";

}

file << endl;

}

file.close();

}

void **read\_file**(){

ifstream infile("test.txt");

ofstream outfile;

outfile.open("data.txt");

unsigned count;

infile >> count;

unsigned i=0;

unsigned border = count/2;

double \*array = new double [count];

while(1){

infile >> array[i];

if(infile.eof()==1){

break;

}

i++;

if(i == count){

for(unsigned j=0;j<count;j++){

if((j>border) || (j%border != 0)){

outfile << array[j] << " ";

}

}

outfile << endl;

i=0;

}

}

infile.close();

outfile.close();

}

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

{

QCoreApplication applications(*argc*, argv);

#ifdef Q\_OS\_WIN32

QTextCodec::setCodecForLocale(QTextCodec::codecForName("IBM 866"));

#endif

#ifdef Q\_OS\_LINUX

QTextCodec::setCodecForLocale(QTextCodec::codecForName("UTF-8"));

#endif

creat\_file();

read\_file();

return applications.exec();

}

Task 2

#include <QCoreApplication>

#include <QTextCodec>

#include <iostream>

using namespace std;

/\* б. Дан файл целых чисел. Нулевых компонент в файле нет. Число отрицательных компонент равно числу положительных.

\* Преобразовать файл таким образом, чтобы сначала были положительные числа, а затем отрицательные.

\* Порядок следования как положительных, так и отрицательных чисел сохранить.\*/

#include <fstream>

void **creat\_file**(){

ofstream file;

file.open("test.txt");

int number;

while(1){

cin >> number;

if(number!=0){

file << number << " ";

} else {

break;

}

}

file.close();

}

void **transfer\_file**(){

ofstream out,out1,out2;

out1.open("data1.txt");

out2.open("data2.txt");

out.open("data.txt");

ifstream file("test.txt");

int number;

while(1){

file >> number;

if(file.eof() == 1){

break;

}

if(number > 0){

out1 << number << " ";

} else {

out2 << number << " ";

}

}

file.close();

out1.close();

out2.close();

ifstream in1("data1.txt");

ifstream in2("data2.txt");

while(1){

in1 >> number;

if(in1.eof() == 1){

break;

}

out << number << " ";

}

while(1){

in2 >> number;

if(in2.eof() == 1){

break;

}

out << number << " ";

}

out.close();

out1.close();

out2.close();

}

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

{

QCoreApplication applications(*argc*, argv);

#ifdef Q\_OS\_WIN32

QTextCodec::setCodecForLocale(QTextCodec::codecForName("IBM 866"));

#endif

#ifdef Q\_OS\_LINUX

QTextCodec::setCodecForLocale(QTextCodec::codecForName("UTF-8"));

#endif

creat\_file();

transfer\_file();

return applications.exec();

}