ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет Вычислительной техники

Кафедра “Вычислительная техника”

**Пояснительная записка**

*по курсовой работе*

*По дисциплине: «*Логика и основы алгоритмизации в инженерных задачах*»*

Выполнил

студент группы

19ВВ3:

Лукьянов Д. Р.

Пенза 2020

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет Вычислительной техники

Кафедра “Вычислительная техника”

“УТВЕРЖДАЮ”

название игры, дата выпуска, разработчик, жанр, издатель, наличие мультиплеера

Зав. кафедрой ВТ

Митрохин М.А.

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_

ЗАДАНИЕ

на курсовое проектирование по курсу

Программирование

Студенту\_ Лукьянову Даниилу Романовичу\_\_\_\_\_\_\_\_\_\_\_\_Группа\_ 19ВВ3\_\_

Тема\_проекта: Унарные и бинарные операции над графами .

Исходные данные (технические требования) на проектирование

Наименование программы: Программа обработки графов различными способами.

Задание: Программа должна выполнять следующие функции:

– Отождествление вершин

– Стягивание ребра

– Расщепление вершин.

– Объединение, пересечение, кольцевая сумма графов.

Расчетная часть

Разработка программы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Графическая часть

Схема данных, схема программы, схема работы системы, иерархическая

структура программы, схема взаимодействия программы.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Экспериментальная часть

Отладка программы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Срок выполнения проекта по разделам

1. В соответствии с графиком выполнения курсового проекта\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания “\_15\_”\_Октября\_2020 г.\_\_\_\_\_\_\_\_\_

Дата защиты проекта “\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Задание получил “\_\_\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020г.

Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оглавление

[Введение 5](#_Toc76317323)

[2 Выбор решения 8](#_Toc76317324)

[2.1 Определение необходимых модулей программы. 8](#_Toc76317325)

[Графическая схема: 9](#_Toc76317326)

[3 Описание разработки программы 10](#_Toc76317327)

[4 Отладка и тестирование 12](#_Toc76317328)

[5 Описание программы 13](#_Toc76317329)

[5.1 Разработка приложения graf.exe 13](#_Toc76317330)

[6 Руководство пользователя 18](#_Toc76317331)

### Введение

К настоящему времени человечеством накоплено поистине гигантское количество информации об объектах и явлениях. Но эта информация не лежит мертвым грузом, она хранится в электронном виде и используется в базах данных.

На вопрос, что такое база данных, информатика дает очень четкий ответ.

Базой данных (БД) называется совокупность материалов, которые систематизированы таким образом, чтобы их было легко найти и обработать с помощью ПК или другой ЭВМ (электронно-вычислительной машины). Под материалами может пониматься все, что угодно: статьи, различные документы, отчеты и т.д.

Система управления базами данных (СУБД) — это совокупность языковых и программных средств, обеспечивающих создание, использование и ведение базы данных.

Исходя из этих определений, нетрудно догадаться, для чего нужна база данных для ПК. Различные БД используются для систематизации и хранения большого количества однотипных документов и быстрого доступа к ним. Помните, когда вы раньше приходили в какую-нибудь крупную библиотеку, сколько времени у библиотекаря занимало нахождение интересующей вас книги? А ведь не всегда можно сразу даже вспомнить, есть ли какой-то материал в картотеке. Теперь эта проблема решена. Достаточно открыть на своем компьютере интересующую БД и набрать в поиске интересующее название. Быстрое нахождение и обработка интересующей информации из списка - вот для чего предназначена база данных и программы, оперирующие различными БД.

История возникновения и развития технологий баз данных может рассматриваться как в широком, так и в узком аспекте.

В широком смысле понятие истории баз данных обобщается до истории любых средств, с помощью которых человечество хранило и обрабатывало данные. В таком контексте упоминаются, например, средства учёта царской казны и налогов в древнем [Шумере](http://ru.wikipedia.org/wiki/%D0%A8%D1%83%D0%BC%D0%B5%D1%80) (4000 г. до [н. э.](http://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D1%88%D0%B0_%D1%8D%D1%80%D0%B0)),  узелковая письменность [инков](http://ru.wikipedia.org/wiki/%D0%98%D0%BD%D0%BA%D0%B8) — [кипу](http://ru.wikipedia.org/wiki/%D0%9A%D0%B8%D0%BF%D1%83), [клинописи](http://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%BD%D0%BE%D0%BF%D0%B8%D1%81%D1%8C" \o "Клинопись),содержащие документы [Ассирийского царства](http://ru.wikipedia.org/wiki/%D0%90%D1%81%D1%81%D0%B8%D1%80%D0%B8%D0%B9%D1%81%D0%BA%D0%BE%D0%B5_%D1%86%D0%B0%D1%80%D1%81%D1%82%D0%B2%D0%BE).

История баз данных в узком смысле рассматривает базы данных в традиционном (современном) понимании. Эта история начинается с [1955 года](http://ru.wikipedia.org/wiki/1955_%D0%B3%D0%BE%D0%B4), когда появилось программируемое оборудование обработки записей. [Программное обеспечение](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) этого времени поддерживало модель обработки записей на основе файлов. Для хранения данных использовались [перфокарты](http://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D1%84%D0%BE%D0%BA%D0%B0%D1%80%D1%82%D0%B0).

Оперативные сетевые базы данных появились в середине [1960-х](http://ru.wikipedia.org/wiki/1960-%D0%B5). Операции над оперативными базами данных обрабатывались в интерактивном режиме с помощью терминалов. Простые индексно-последовательные организации записей быстро развились к более мощной модели записей, ориентированной на наборы.

В это же время в сообществе баз данных [COBOL](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B1%D0%BE%D0%BB) была проработана концепция [схем баз данных](http://ru.wikipedia.org/wiki/%D0%A1%D1%85%D0%B5%D0%BC%D0%B0_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) и концепция независимости данных.

Следующий важный этап связан с появлением в начале [1970-х](http://ru.wikipedia.org/wiki/1970-%D0%B5) [реляционной модели данных](http://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), благодаря работам [Эдгара Ф. Кодда](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4%D0%B4,_%D0%AD%D0%B4%D0%B3%D0%B0%D1%80). Работы Кодда открыли путь к тесной связи прикладной технологии баз данных с математикой и логикой.

**1 Постановка задачи**

Необходимо разработать программу – базу данных «Компьютерные игры». Программа содержит следующие поля: Название, жанр, год, тематика, цена на рынке и производитель.

Программа должна позволить создать и сохранить базу данных, добавлять и удалять записи, осуществлять поиск и сортировку элементов. Для реализации этих требований нужно разработать интуитивно понятный интерфейс пользователя и определить названия пунктов в меню. Необходимо разработать алгоритм функционирования меню и обработки событий клавиатуры в консоли. Важно продумать структуру работы программы и механизм взаимодействия входящих в нее функций.

**Многомодульность программы**. Необходимо поделить программу на логические модули. Это повысит расширяемость, упростит возможность редактирования программы и исправления возникающих ошибок при отладке или тестировании программы.

Использование сложных типов данных – массивов, структур, файлов. Необходимо изучить базовые и сложные типы данных. Это необходимо для структуризации программы. Важно изучить работу с файлами. Это необходимо для возможности сохранения базы данных в файл и дальнейшего считывания из него.

**Устройство ввода информации** – клавиатура или мышь. Необходимо научиться анализировать события, возникающие от работы с клавиатурой или мышью. Необходимо однозначно идентифицировать и выполнять те или иные действия в зависимости от действий пользователя.

Пользовательский интерфейс должен быть построен на основе меню и панели инструментов. Необходимо изучить приемы программирования меню. Это нужно для создания интуитивно понятного интерфейса

### 2 Выбор решения

### 2.1 Определение необходимых модулей программы.

Разработанная программа состоит из 7 модулей:

1. mas1vyvod
2. mas2vyvod
3. otozhd
4. ras
5. kon
6. diz
7. xili

Рисунок 1 – структура модулей программы

Модуль mas1vyvod выводит граф mas1.

Модуль mas2vyvod выводит граф mas2.

Модуль otozhd выполняет операцию отождествления вершин.

Модуль ras выполняет операцию расщепления вершин.

Модуль kon выполняет операцию пересечения графов.

Модуль diz выполняет операцию объединения графов.

Модуль zili выполняет операцию исключающего или графов.

2.2 Определение структуры файла базы данных.

Была выбрана следующая структура данных:

Массив вершин

Графы реализованы в виде массива вершин. Программа умеет делать следующее:

1. выполнять операцию отождествления вершин.
2. выполнять операцию расщепления вершин.
3. выполнять операцию пересечения графов.
4. выполнять операцию объединения графов.
5. выполнять операцию исключающего или графов.

На рисунке 2 представлена схема данных:

Пользователь

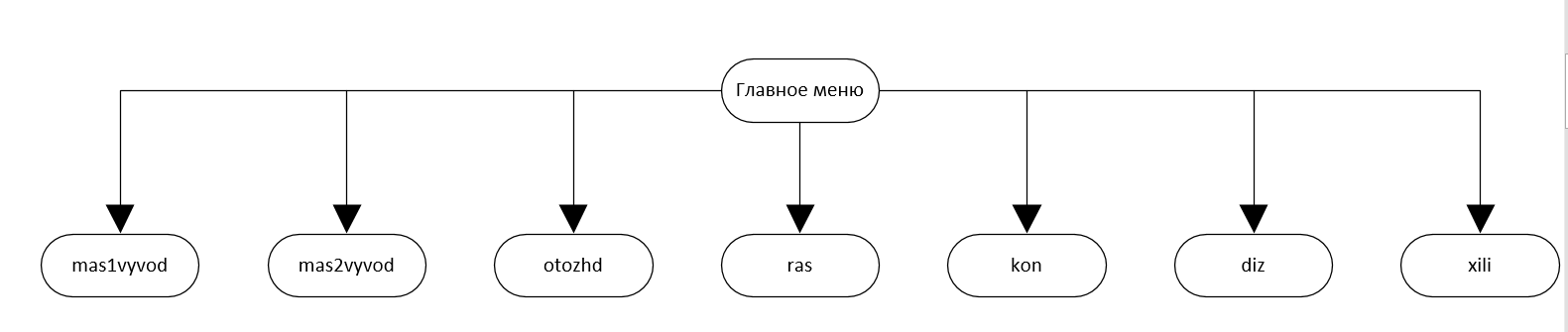
Операционная система

Программа

Устройство ввода

Дисплей

### Графическая схема:



### 3 Описание разработки программы

Для написания данной программы будет использован язык программирования Си. Этот язык является распространённым языком программирования. При разработке языка Си был принят компромисс между низким уровням языка ассемблера и высоким уровнем других языков. В языке Си предусмотрено много операций, непосредственно исполняемых большинством микропроцессоров; в то же время он даёт программисту возможность выражать свои алгоритмы и данные наиболее подходящими средствами. Язык Си обеспечивает возможности структурирования данных. Он позволяет разрабатывать большие, но структурированные программы, предоставляя возможность отдельной разработки подпрограмм.

Целью создателей языка Си была разработка мобильного языка, который можно было бы использовать для разработки системного программного обеспечения. Программы, написанные на языке высокого уровня, мобильны, поскольку язык должен быть одним и тем же независимо от того, на каком компьютере и в какой операционной системе он используется. Дополнительное свойство языка Си, называемое условной компиляцией, позволяет программисту изолировать машинно-зависимые операторы и контролировать их компиляцию в другой среде. Это дополнительно повышает мобильность программного обеспечения, написанного на языке Си.

В качестве среды программирования был выбран программный продукт Microsoft Visual Studio 2012.

Язык Си был разработан американцем *Деннисом Ритчи* в исследовательском центре Computer Science Research Center of Bell Laboratories корпорации AT&T в 1972 г. Первоначальная реализация Си была выполнена на ЭВМ PDP-11 фирмы DEC для создания операционной системы UNIX. Позже он был перенесен в среду многих операционных систем, обособился и существует независимо от любой из них.

Первое описание языка Си дал его автор Деннис Ритчи совместно с Брайном Керниганом в книге «Язык программирования Си». Однако, описание не было строгим и содержало ряд неоднозначных моментов.

В 1983 г. при Американском Институте Национальных Стандартов (American National Standart Institute — ANSI) был создан комитет по стандартизации языка Си. В 1989 г. был утверждён окончательный вариант стандарта. Однако на сегодняшний день большинство реализаций языка Си не поддерживают стандарт в полном объёме.

При запуске программы на экран выводятся действия, которые можно произвести (добавить игру, вывести список, читать с файла и выход из программы).

Для отождествления вершин нужно нажать на кнопку “F1” и последовать дальнейшим инструкциям (для стягивания ребра нужно выбрать смежные вершины).

Для расщепления вершин нужно нажать на кнопку “F2” и последовать дальнейшим инструкциям.

Для конъюнкции графов нужно нажать “F7”.

Для дизъюнкции графов нужно нажать “F8”.

Для исключающего или графов нужно нажать “F9”.

### 4 Отладка и тестирование

В качестве среды разработки была выбрана программа Visual Studio 2019. Программа предоставляет все средства необходимые при разработке и отладке разрабатываемых модулей и программ. Для отладки использовались такие инструменты как точка остановки, выполнение кода по шагам, анализ содержимого локальных и глобальных переменных, анализ содержимого памяти.

Тестирование проводилось в рабочем порядке, в процессе разработки, после завершения написания программы. В ходе тестирования было выявлено и исправлено множество проблем, связанных с обработкой структуры, системой поиска и Ассемблером.

### 5 Описание программы

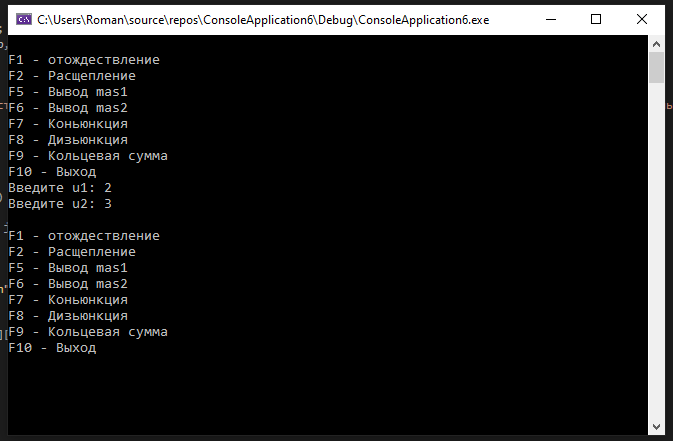
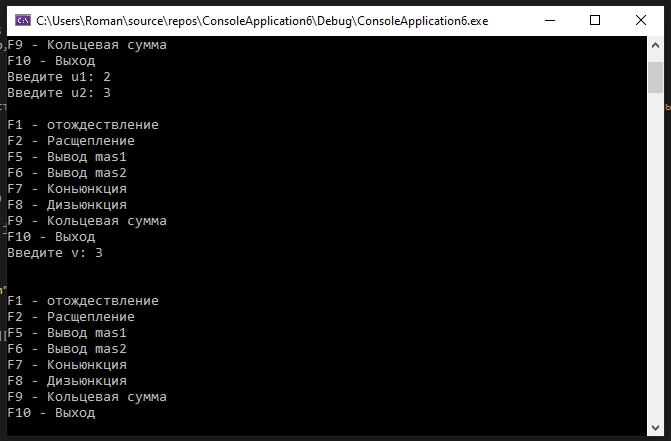
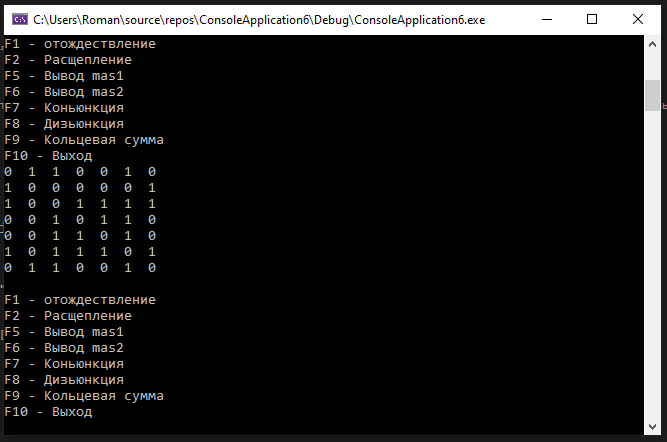
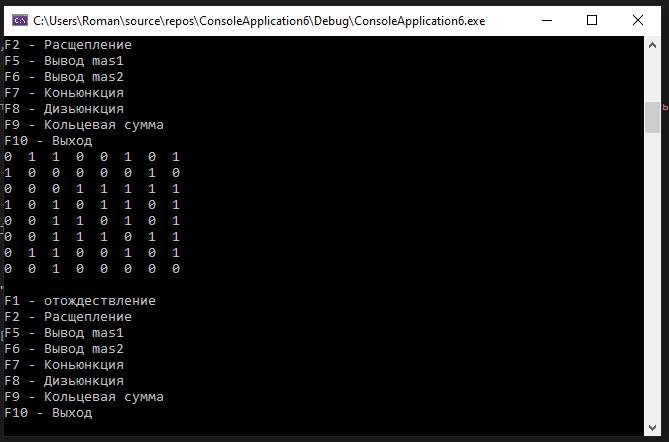
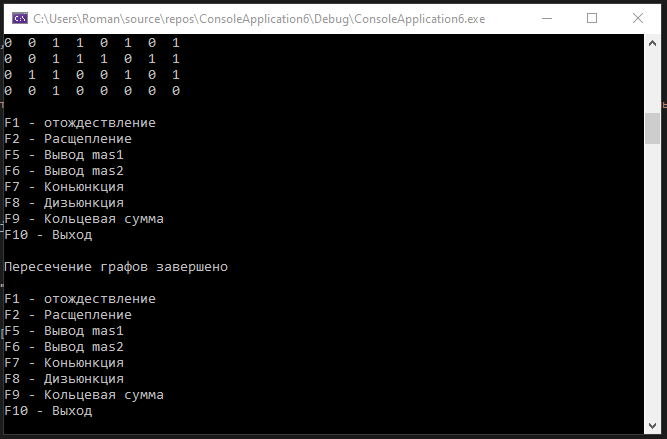
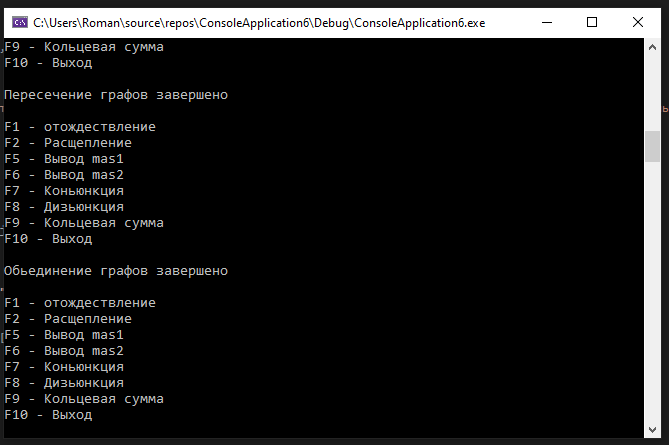
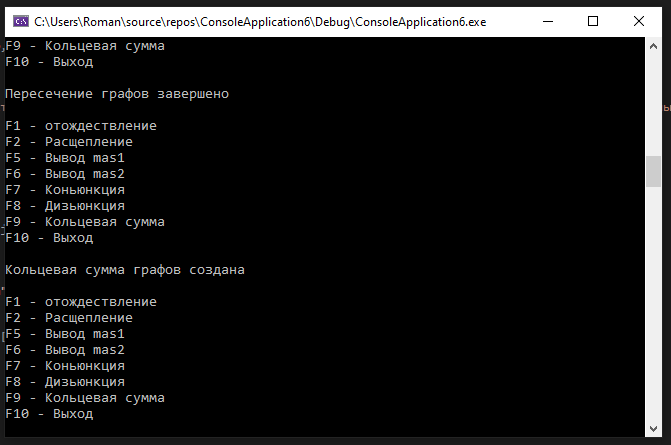
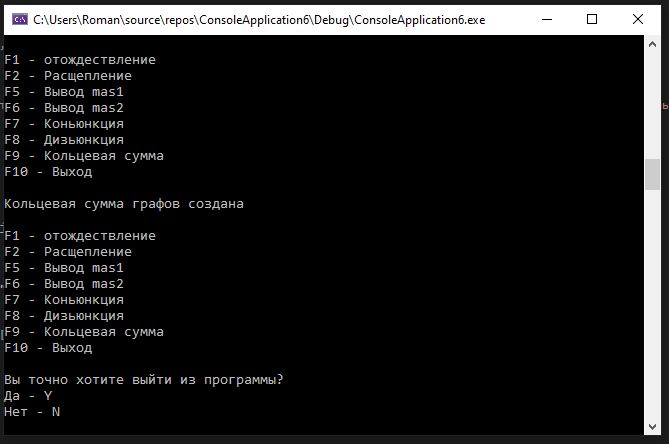
### 5.1 Разработка приложения graf.exe

Разработанная программа состоит из нескольких модулей. Такое разбиение позволяет более эффективно работать с кодом программы, быстрее находить и исправлять возникшие ошибки.

Приложение graf.exeявляется основным модулем программы. Для реализации интерфейса был разработан модуль Menu.cpp. В нем содержится несколько функций, необходимых для работы меню, вывода его на экран и выбора необходимого действия. Также вспомогательные меню других функций содержатся в других модулях. Алгоритмы их работы одинаковы.

При запуске программы на экран выводится заставка-приветствие, после чего программа готова к выполнению своих функций. Происходит ожидание нажатия клавиши, и, в зависимости от нажатой клавиши, передается управление необходимой функции.

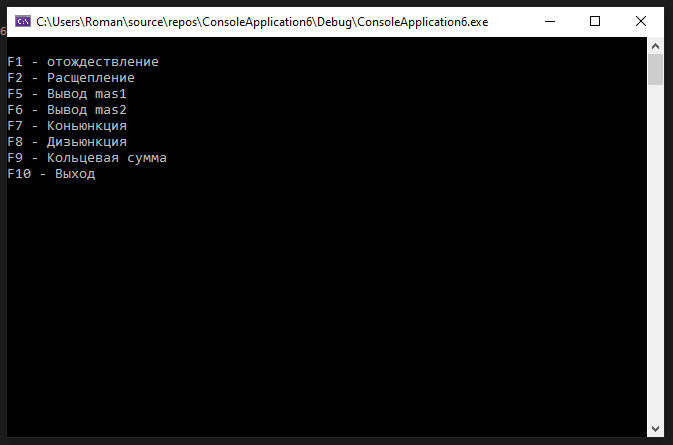
Все пункты меню:



### 6 Руководство пользователя

Программа предназначена для обработки графов. Программа создает массивы mas1 и mas2, над которыми и будут производиться действия

При запуске программы на экран выводятся действия, которые можно произвести (добавить игру, вывести список, читать с файла и выход из программы).



Далее выбираем действия, которые нужно произвести. Результат сохраняется в mas2.

**Листинг программы**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <stdio.h>

#include <conio.h>

#include <locale.h>

#include <windows.h>

using namespace std;

int mas1[7][7], mas2[8][8];

int button, sch, i, j, a, b, u1, u2, v;

void menu()

{

cout << "\nF1 - отождествление\nF2 - Расщепление\nF5 - Вывод mas1\nF6 - Вывод mas2\nF7 - Коньюнкция\nF8 - Дизьюнкция\nF9 - Кольцевая сумма\nF10 - Выход\n";

}

void mas1vyvod()

{

sch = 0;

for (i = 0; i < 7; i++)

{

for (j = 0; j < 7; j++)

{

if (sch == 7)

{

cout << "\n";

sch = 0;

}

cout << mas1[i][j] << " ";

sch = sch + 1;

}

}

cout << "\n";

menu();

}

void mas2vyvod()

{

sch = 0;

for (i = 0; i < 8; i++)

{

for (j = 0; j < 8; j++)

{

if (sch == 8)

{

cout << "\n";

sch = 0;

}

cout << mas2[i][j] << " ";

sch = sch + 1;

}

}

cout << "\n";

menu();

}

void otozhd()

{

cout << "Введите u1: ";

cin >> u1;

cout << "Введите u2: ";

cin >> u2;

for (i = 0; i < 7; i++)

{

b = 0;

if ((i == u1) || (i == u2)) continue;

for (j = 0; j < 7; j++)

{

if ((j == u1) || (j == u2)) continue;

mas2[a][b] = mas1[i][j];

b++;

}

mas2[a][b] = mas1[u1][j] | mas1[u2][j];

mas2[b][a] = mas2[a][b];

a++;

}

menu();

}

void ras()

{

cout << "Введите v: ";

cin >> v;

if ((v == 0) || (v < 0) || (v > 8))

{

cout << "\nНевозможное значение\n";

menu();

return;

}

else

{

v = v - 1;

}

for (i = 0; i < 7; i++)

{

for (j = 1; j < 7; j++)

{

mas2[i][j] = mas1[i][j];

if (j == v)

{

mas2[7][j] = mas1[i][j];

mas2[i][7] = mas1[i][j];

}

}

}

mas2[7][v] = 1;

mas2[v][7] = 1;

cout << "\n";

menu();

}

void kon()

{

for (i = 0; i < 7; i++)

{

for (j = 0; j < 7; j++)

{

mas1[i][j] and mas2[i][j];

}

}

cout << "\nПересечение графов завершено\n";

menu();

}

void diz()

{

for (i = 0; i < 7; i++)

{

for (j = 0; j < 7; j++)

{

mas1[i][j] or mas2[i][j];

}

}

cout << "\nОбьединение графов завершено\n";

menu();

}

void xili()

{

for (i = 0; i < 7; i++)

{

for (j = 0; j < 7; j++)

{

mas1[i][j] xor mas2[i][j];

}

}

cout << "\nКольцевая сумма графов создана\n";

menu();

}

int main()

{

setlocale(LC\_ALL, "Rus");

for (i = 0; i < 7; i++)

{

for (j = i; j < 7; j++)

{

if (i == j)

{

mas1[i][j] = 0;

}

else

{

mas1[i][j] = rand() % 2;

mas1[j][i] = mas1[i][j];

}

}

}

menu();

while ((button != 89) || (button != 121))

{

switch (button = \_getch())

{

menu();

case 59:

otozhd();

break;

case 60:

ras();

break;

case 63:

mas1vyvod();

break;

case 64:

mas2vyvod();

break;

case 65:

kon();

break;

case 66:

diz();

break;

case 67:

xili();

break;

case 68: //ВЫХОД

cout << "\nВы точно хотите выйти из программы?\nДа - Y\nНет - N\n";

button = \_getch();

if ((button == 89) || (button == 121))

{

exit(0);

}

else

{

if ((button == 78) || (button == 110))

{

menu();

break;

}

}

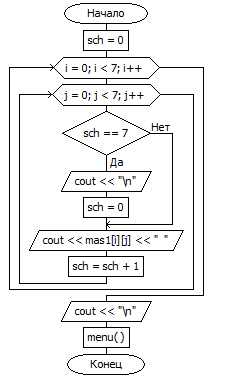
}

}

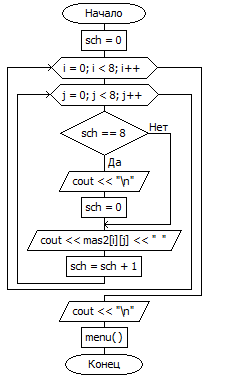
}

**Блок-схемы модулей с алгоритмами**

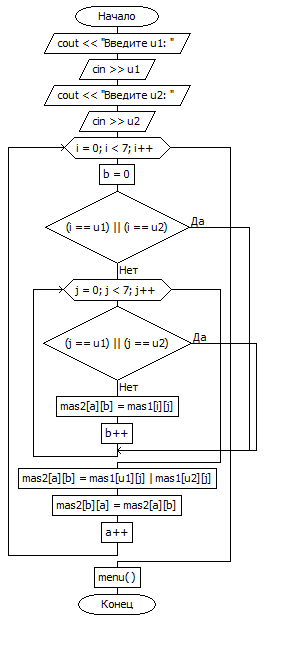
**mas1vyvod()**

****

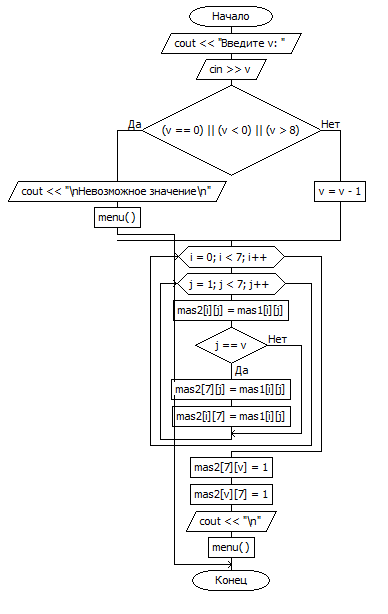
**mas2vyvod()**

****

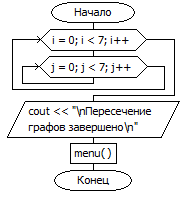
**otozhd()**

****

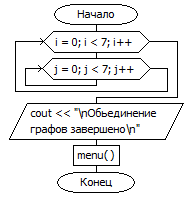
**ras()**

****

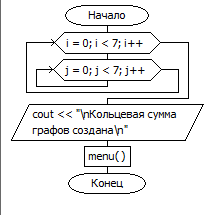
**kon()**

****

**diz()**

****

**xili()**

****

**Заключение**

При выполнении данной курсовой работы были получены навыки разработки многомодульных программ. Были освоены приемы создания пользовательского интерфейса, изучены функции работы с консолью, способы обработки событий с клавиатуры. Усвоены механизмы реализации меню. Изучены функции работы с файлами. Получены базовые навыки программирования на языке Си. Изучены основные возможности среды программирования Microsoft Visual Studio 2012. Получены навыки отладки и тестирования программ.

В рамках выполнения курсовой работы была написана база данных. Программа представляет небольшой, но достаточный список возможностей.

В дальнейшем программу можно улучшить, изменив интерфейс, добавив псевдографику, шифрование данных и отдельные аккаунты.