

ЗАДАНИЕ на учебную практику

по теме Проектирование и реализация программного обеспечения с использованием
структурного и объектного подходов

Студент группы ИУ6-23Б

Залыгин Вячеслав Константинович
(Фамилия, имя, отчество)

Тип практики Проектно-технологическая практика

Название предприятия НУК ИУ МГТУ им. Н.Э. Баумана

Техническое задание:

Задание 1. Создание программной системы на Object Pascal

Выполнить объектную декомпозицию, разработать формы интерфейса, диаграмму состояний интерфейса, диаграммы классов интерфейсной и предметной областей, диаграмму последовательности действий одной из реализуемых операций. Разработать, протестировать и отладить программу.

Сведения о студентах включают: фамилию, имя, индекс группы, рейтинг (от 0 до 100). Программа должна в интерактивном режиме формировать файл, добавлять и удалять данные, а также воспринимать каждый из перечисленных запросов и давать на него ответ.

1. Выяснить, имеются ли в институте однофамильцы. Если имеются, показать их фамилии и имена.
2. Выяснить, в каких группах обучается количество студентов более заданного.
3. Получить список студентов с указанием группы, рейтинг которых не меньше зачетного (60).
4. Построить гистограмму, показывающую средний рейтинг по каждой группе.

Задание 2. Создание программной системы с элементарным интерфейсом консольного режима на C++

Выполнить структурную декомпозицию, разработать структурную схему, содержащую не менее 3 подпрограмм, и алгоритмы этих подпрограмм. Реализовать на C++ в консольном режиме. Предусмотреть примитивный интерфейс типа меню, позволяющий выбирать нужную подпрограмму.

Разработать программу, которая реализует операции над матрицами. Реализовать следующие операции: ввод матрицы, умножение матрицы на число, получение верней

треугольной и нижней треугольной матриц из заданной матрицы, а также вывод результатов операций на экран.

Задание 3. Создание программной системы с Qt интерфейсом на C++

Выполнить объектную декомпозицию, разработать формы интерфейса, диаграмму состояний интерфейса, диаграммы классов интерфейсной и предметной областей, диаграмму последовательности действий одной из реализуемых операций. Разработать, протестировать и отладить программу в среде Visual Studio или QT Creator.

Сведения о студентах включают: фамилию, имя, индекс группы, рейтинг (от 0 до 100). Программа должна в интерактивном режиме формировать файл, добавлять и удалять данные, а также воспринимать каждый из перечисленных запросов и давать на него ответ.

1. Выяснить, имеются ли в институте однофамильцы. Если имеются, показать их фамилии и имена.
2. Выяснить, в каких группах обучается количество студентов более заданного.
3. Получить список студентов с указанием группы, рейтинг которых не меньше заданного (60).
4. Построить гистограмму, показывающую средний рейтинг по каждой группе.

Оформление отчета по практике:

Отчет на 25-35 листах формата А4 должен включать титульный лист, задание (печатать с двух сторон), оглавление, введение, три главы, заключение и список использованных источников. Отдельная глава по каждому заданию должна содержать анализ задания, требуемые чертежи, текст программы, результаты тестирования и выводы.

Дата выдачи задания « 07 » февраля 2023 г.

Руководитель практики

(Подпись, дата)

А.М. Минтаева

(И.О. Фамилия)

Студент

(Подпись, дата)

В.К. Залыгин

(И.О. Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

ОТЧЕТ ОБ УЧЕБНОЙ ПРАКТИКЕ

Тип практики Проектно-технологическая практика

Название предприятия НУК ИУ МГТУ им. Н.Э. Баумана

Студент группы ИУ6-23Б

26.03.2023

(Подпись, дата)

В.К. Залыгин

(И.О. Фамилия)

Руководитель практики

22.05.2023

(Подпись, дата)

А.М. Минитаева

(И.О. Фамилия)

Оценка отлично

2023 г.



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

ОТЧЕТ ОБ УЧЕБНОЙ ПРАКТИКЕ

Тип практики Проектно-технологическая практика

Название предприятия НУК ИУ МГТУ им. Н.Э. Баумана

Студент группы ИУ6-23Б

26.03.2023

(Подпись, дата)

В.К. Залыгин

(И.О. Фамилия)

Руководитель практики

(Подпись, дата)

А.М. Минитаева

(И.О. Фамилия)

Оценка _____

2023 г.

Задание 1. Создание программной системы на Object Pascal

Выполнить объектную декомпозицию, разработать формы интерфейса, диаграмму состояний интерфейса, диаграммы классов интерфейсной и предметной областей, диаграмму последовательности действий одной из реализуемых операций. Разработать, протестировать и отладить программу.

Сведения о студентах включают: фамилию, имя, индекс группы, рейтинг (от 0 до 100). Программа должна в интерактивном режиме формировать файл, добавлять и удалять данные, а также воспринимать каждый из перечисленных запросов и давать на него ответ.

1. Выяснить, имеются ли в институте однофамильцы. Если имеются, показать их фамилии и имена.
2. Выяснить, в каких группах обучается количество студентов более заданного.
3. Получить список студентов с указанием группы, рейтинг которых не меньше зачетного (60).
4. Построить гистограмму, показывающую средний рейтинг по каждой группе.

Цель работы

Получение практических навыков разработки прикладного приложения с графическим интерфейсом с использованием языка программирования ObjectPascal и библиотеки компонентов LCL.

Интерфейс приложения

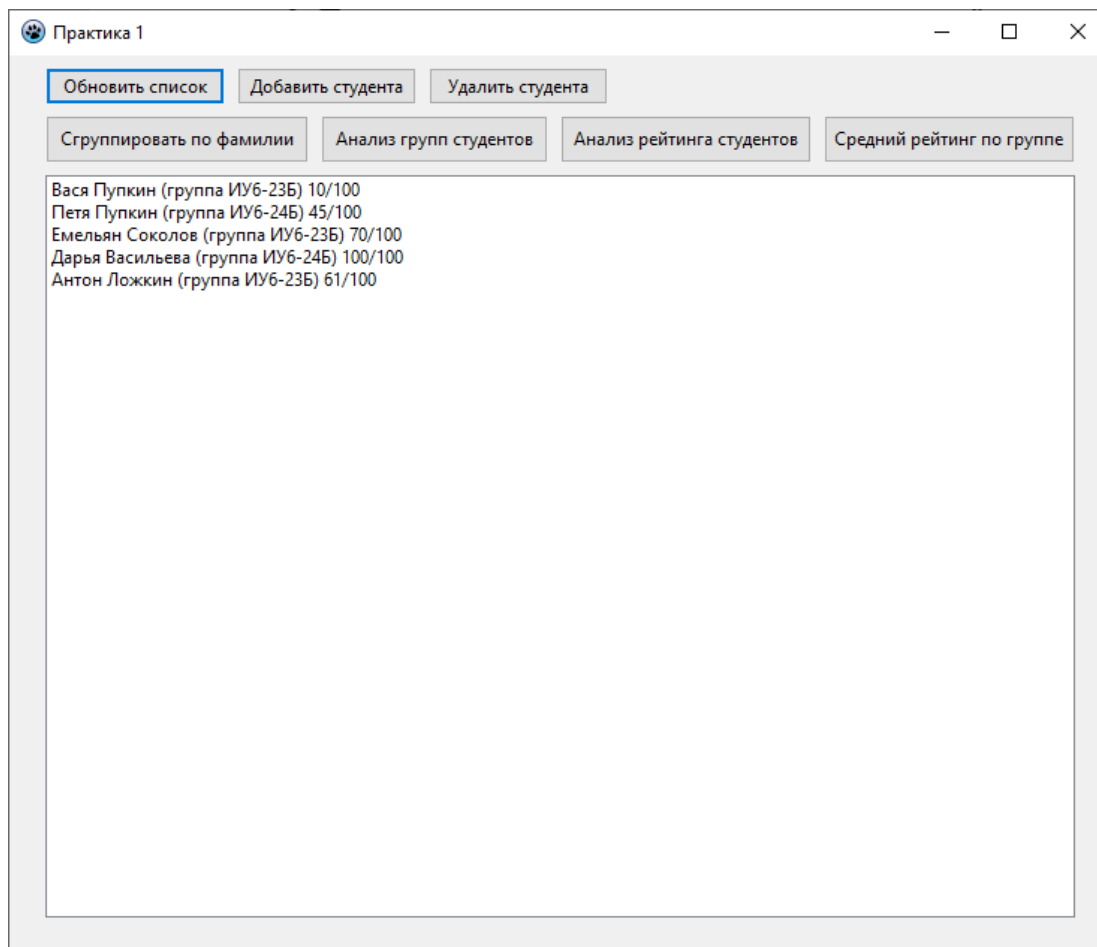


Рисунок 1 - главное меню программы

The screenshot shows a dialog box titled 'Новый студент'. It contains four text input fields stacked vertically, labeled 'Имя', 'Фамилия', 'Группа', and 'Рейтинг (от 0 до 100)'. At the bottom of the dialog is a button labeled 'Добавить студента', which is highlighted with a blue border.

Рисунок 2 - форма добавления нового студента

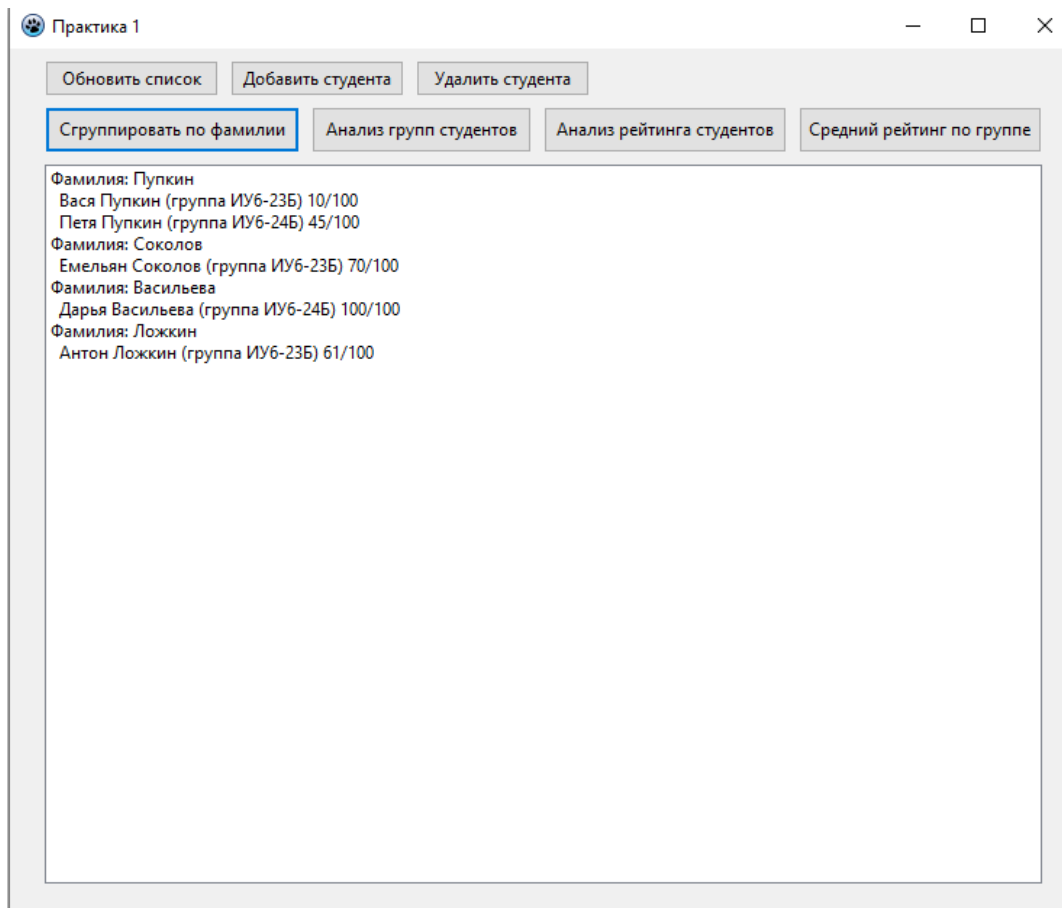


Рисунок 3 - режим группировки по фамилии

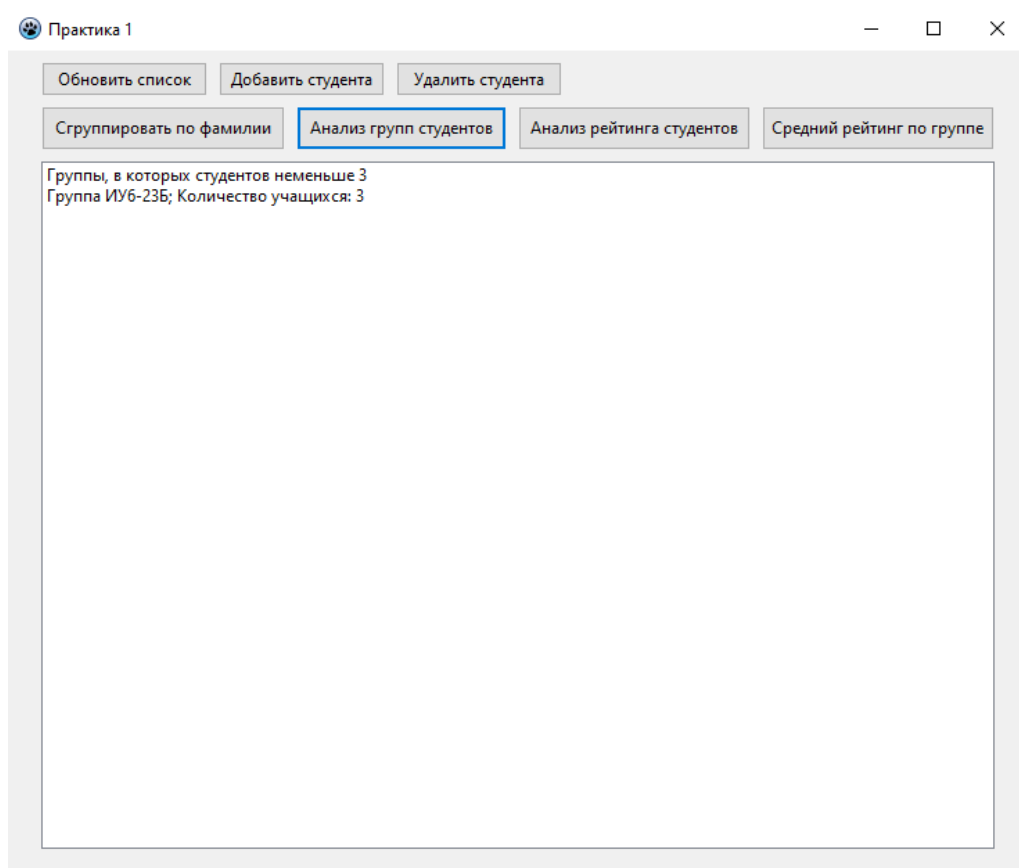


Рисунок 4 - группировка по количеству студентов в группе

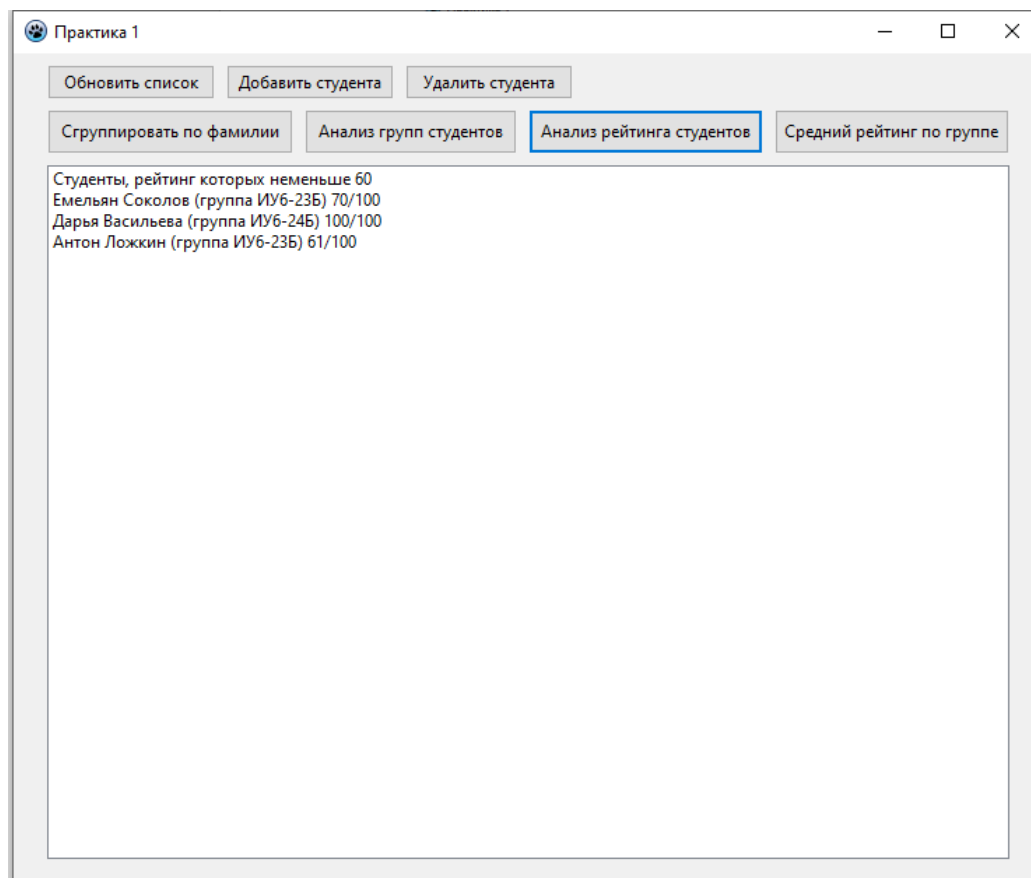


Рисунок 5 - фильтр по количеству баллов

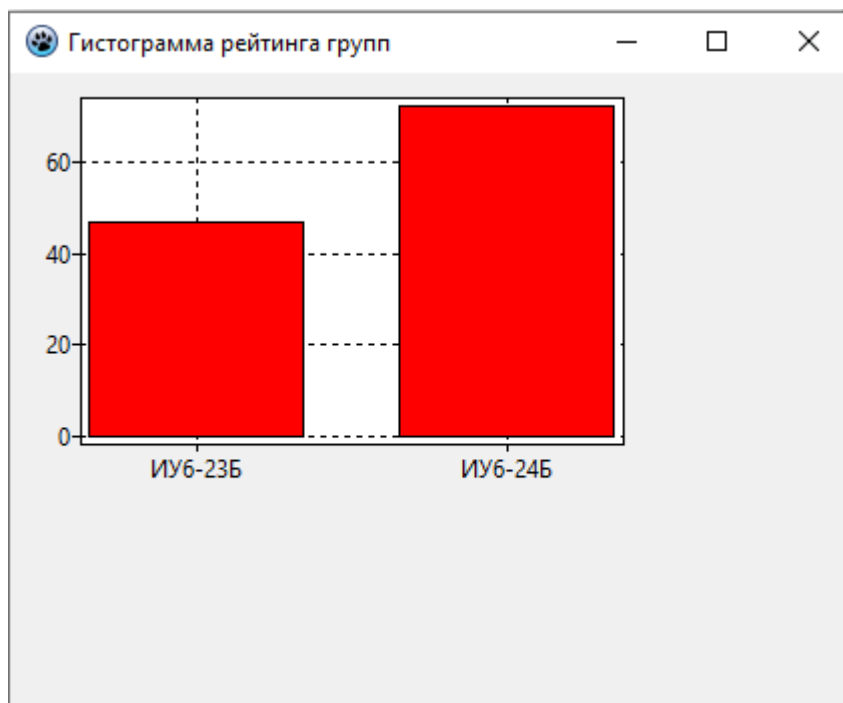


Рисунок 6 - гистограмма среднего рейтинга групп

Проект приложения

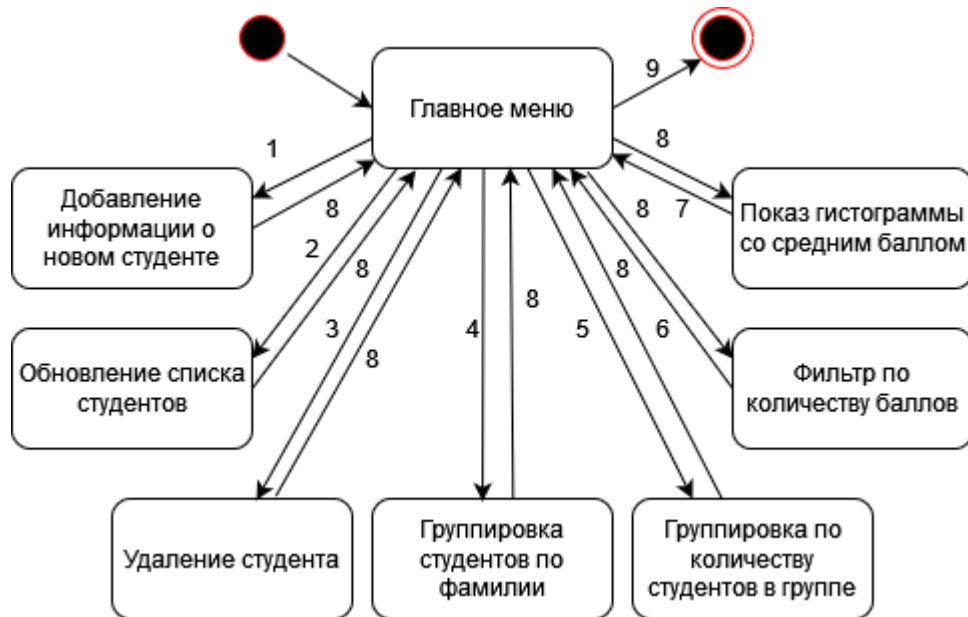


Рисунок 7 - диаграмма состояний

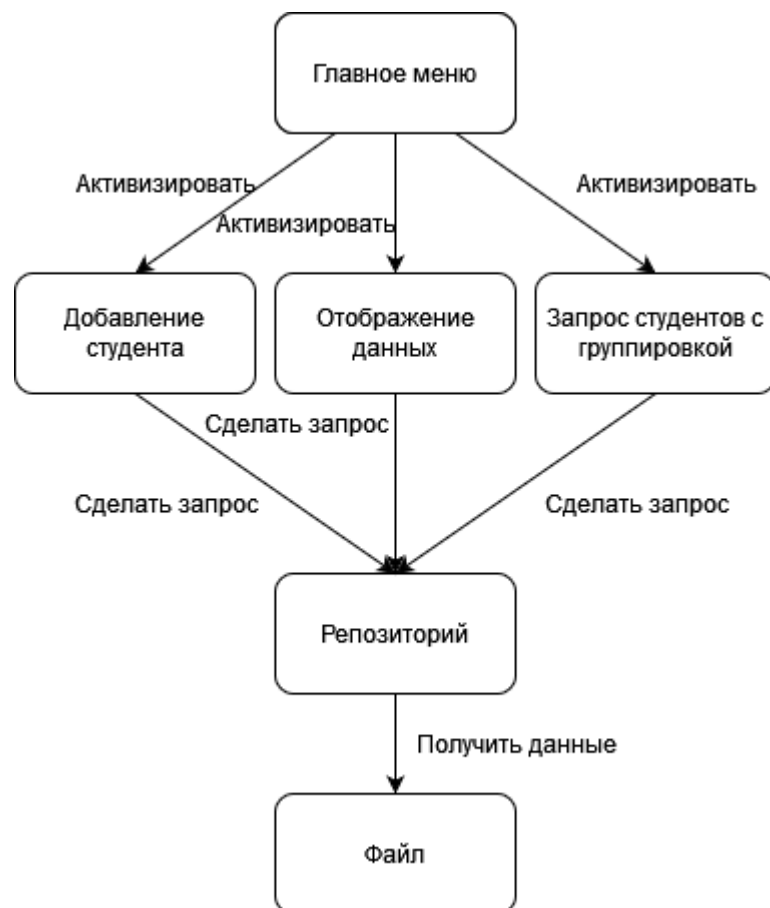


Рисунок 8 - объектная декомпозиция

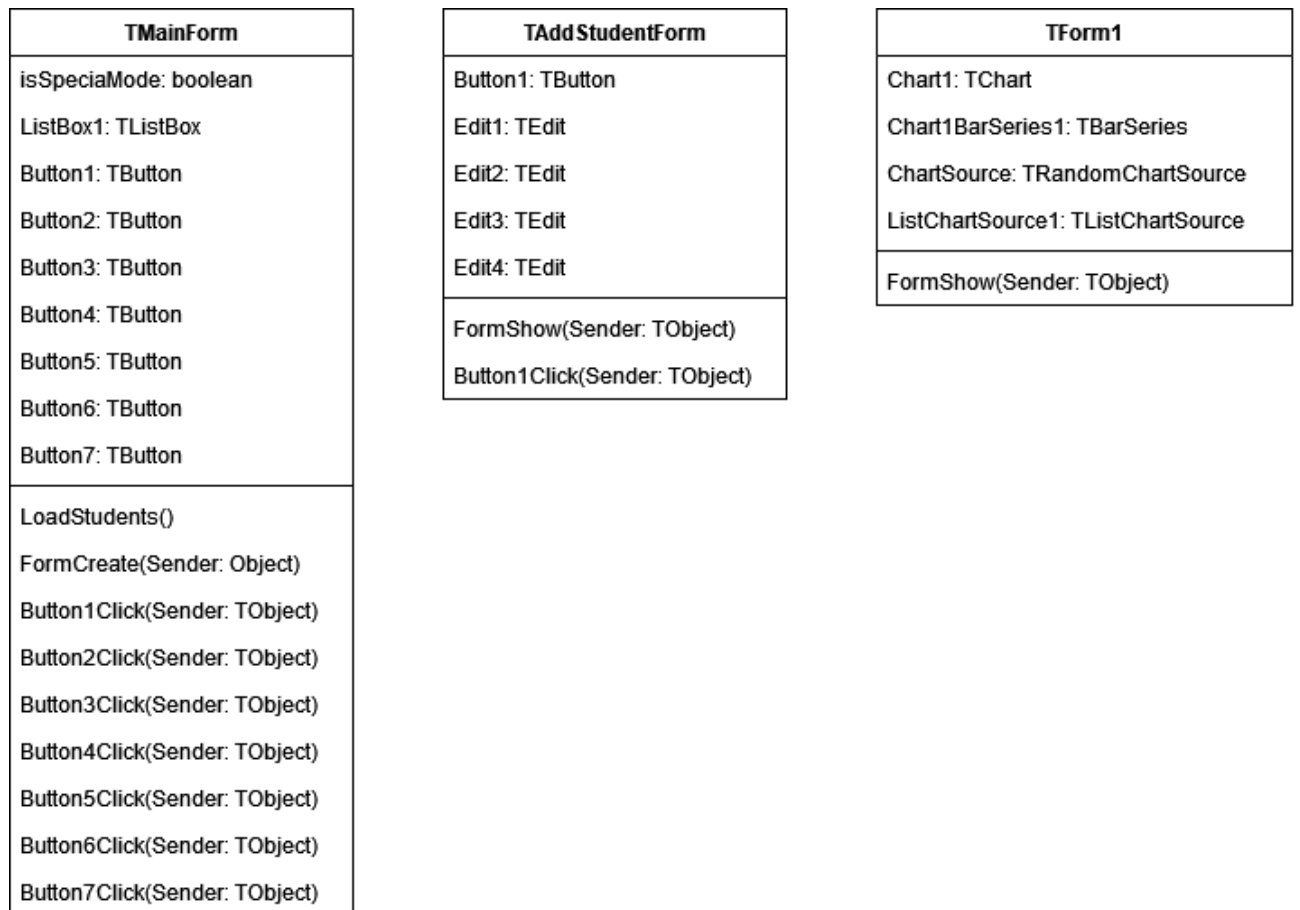


Рисунок 9 - объектная декомпозиция интерфейса

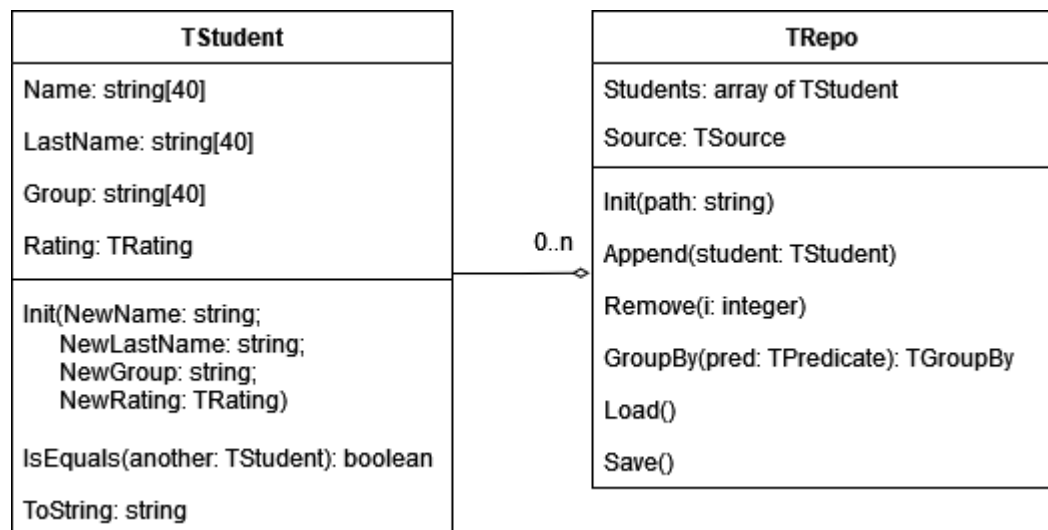


Рисунок 10 - объектная декомпозиция предметной области

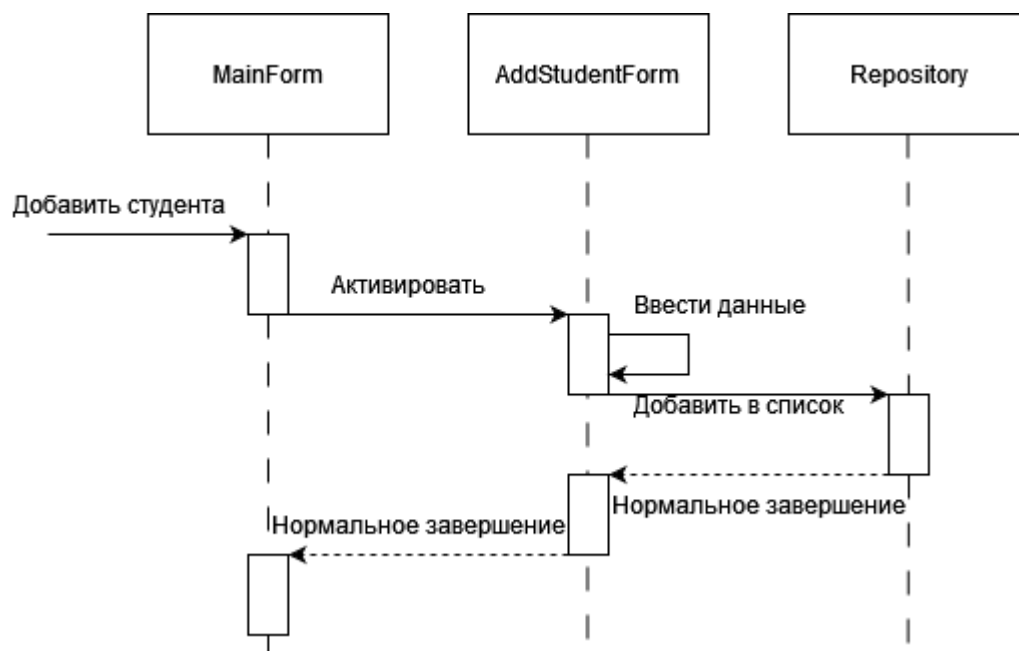


Рисунок 11 - диаграмма последовательности действий при добавлении нового студента

Код приложения

```

1  unit domain;
.
.  {$mode ObjFPC}{$H+}
.
5  interface
.
.  uses
.  [
.      Classes, SysUtils;
.  ]
.
10 type
.    TRating = 0..100;
.
.  TStudent = object
.  public
15      Name: string[40];
.      LastName: string[40];
.      Group: string[40];
.      Rating: TRating;
.  public
20      constructor Init(
.          NewName: string;
.          NewLastName: string;
.          NewGroup: string;
.          NewRating: TRating);
25      function IsEquals(another: TStudent): boolean;
.      function ToString: string;
.  end;
.
.  TGroupBy = array of array of TStudent;
30  TPredicate = function(a, b: TStudent): boolean;
.
.  function isNamesakes(a, b: TStudent): boolean;
.  function isGroupmates(a, b: TStudent): boolean;

```

Рисунок 12 - код модуля домена


```

1  unit repository;
.
.  {$mode delphi}{$H+}
.
5  interface
.
.  uses
.  [
.    Classes, SysUtils, Domain;
.  ]
.
10 type
.   TSource = file of TStudent;
.
.   TRepo = object
.   [
.   [
15     Students: array of TStudent;
.     Source: TSource;
.   [
.     public
.     constructor Init(path: string);
.     procedure Append(student: TStudent);
20     procedure Remove(i: integer);
.     function GroupBy(pred: TPredicate): TGroupBy;
.     procedure Load;
23     procedure Save;
.   end;
.
25
.  var Repo: TRepo;
.

```

Рисунок 14 - код модуля репозитория

```

.   implementation
.
30  constructor TRepo.Init(path: string);
.   begin
.       if not FileExists(path) then
.           FileCreate(path);
.           AssignFile(Source, path);
35  end;
.
.   procedure TRepo.Append(student: TStudent);
.   begin
.       SetLength(Students, Length(Students)+1);
40  Students[Length(Students)-1] := student;
.       Save;
.   end;
.
.   procedure TRepo.Remove(i: integer);
45  begin
.   if (Length(Students) > 0) and (i > -1) then begin
.       Students[i] := Students[Length(Students)-1];
.       SetLength(Students, Length(Students)-1);
.   end;
50  Save;
.   end;
.
.   function TRepo.GroupBy(pred: TPredicate): TGroupBy;
.   var i, j: integer;
55  found: boolean;
.   begin
.   for i := 0 to High(Students) do begin
.       found := false;
.       j := 0;
60  while (j < Length(Result)) and not found do begin
.   if pred(Students[i], Result[j][0]) then begin
.       SetLength(Result[j], Length(Result[j]) + 1);
.       Result[j][Length(Result[j])-1] := Students[i];
.       found := true;
65  end;
.       j += 1;
.   end;
.   if not found then begin
.       j := Length(Result);
70  SetLength(Result, j+1);
.       SetLength(Result[j], 1);
.       Result[j][0] := Students[i];
.   end;
.   end;
75  end;

```

Рисунок 15 - код модуля репозитория


```

. procedure TRepo.Load;
.   var length, i: integer;
.   begin
80     reset(Source);
.     length := FileSize(Source);
.     SetLength(Students, length);
.     i := 0;
.   while not eof(Source) do begin
85     Read(Source, Students[i]);
.     i += 1;
.   end;
.   close(Source);
.   end;
90
. procedure TRepo.Save;
.   var i: integer;
.   begin
.     rewrite(Source);
95   for i := 0 to High(Students) do begin
.     Write(Source, Students[i]);
.   end;
.   close(Source);
.   end;
100
. end.

```

Рисунок 16 - код модуля репозитория

```

1  unit main;
.
.  {$mode delphi}{$H+}
.
5  interface
.
.  uses
8    Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls,
.    Repository, Domain, AddStudent, Gist;
10
.  type
.
.  { TMainForm }
.
15 TMainForm = class(TForm)
.    Button1: TButton;
.    Button2: TButton;
.    Button3: TButton;
.    Button4: TButton;
20    Button5: TButton;
.    Button6: TButton;
.    Button7: TButton;
.    ListBox1: TListBox;
.    procedure Button4Click(Sender: TObject);
25    procedure Button5Click(Sender: TObject);
.    procedure Button6Click(Sender: TObject);
.    procedure Button7Click(Sender: TObject);
.    procedure LoadStudents;
.    procedure Button1Click(Sender: TObject);
30    procedure Button2Click(Sender: TObject);
.    procedure Button3Click(Sender: TObject);
.    procedure FormCreate(Sender: TObject);
.  private
.    isSpecialMode: boolean;
35  public
.
.    end;
.
.  var
40    MainForm: TMainForm;
.

```

Рисунок 17 - код модуля main

```

.   implementation
.
.   {$R *.lfm}
45
.   { TMainForm }
47
.   procedure TMainForm.LoadStudents;
.   var
50     i: integer;
.   begin
.     ListBox1.Clear;
.     for i := 0 to High(Repo.Students) do
.     begin
55       ListBox1.Items.Add(Repo.Students[i].ToString);
.     end;
.     isSpecialMode := False;
.   end;
.
.   procedure TMainForm.FormCreate(Sender: TObject);
.   begin
.     Repo.Init('students.dat');
.     Repo.Load;
.   end;
65
.   procedure TMainForm.Button1Click(Sender: TObject);
.   begin
.     AddStudentForm.Show;
.   end;
70
.   procedure TMainForm.Button2Click(Sender: TObject);
.   begin
.     if not isSpecialMode then
.     begin
75       Repo.Remove(ListBox1.ItemIndex);
.       LoadStudents;
.     end;
.   end;
.
.   procedure TMainForm.Button3Click(Sender: TObject);
80   begin
.     LoadStudents;
.   end;

```

Рисунок 18 - код модуля main

```

85 procedure TMainForm.Button4Click(Sender: TObject);
.   var
.     namesakes: TGroupBy;
.     i, j: integer;
.   begin
90     namesakes := Repo.GroupBy(isNamesakes);
.     isSpecialMode := True;
.     ListBox1.Clear;
.     for i := 0 to High(namesakes) do
.       begin
95         ListBox1.Items.Add('Фамилия: ' + namesakes[i][0].LastName);
.         for j := 0 to High(namesakes[i]) do
.           begin
.             ListBox1.Items.Add(' ' + namesakes[i][j].ToString);
.           end;
100        end;
.      end;
.    end;

.   procedure TMainForm.Button5Click(Sender: TObject);
.   var
105     groupmates: TGroupBy;
.     StudentsAmountStr: string;
.     i, StudentsAmountInt: integer;
.   begin
.     InputQuery('Фильтр',
110       'Порог фильтрации групп по количеству студентов (>=)',
.       StudentsAmountStr);
.     StudentsAmountInt := StrToInt(StudentsAmountStr);
.     groupmates := Repo.GroupBy(isGroupmates);
.     ListBox1.Clear;
115     ListBox1.Items.Add('Группы, в которых студентов неменьше ' + StudentsAmountStr);
.     isSpecialMode := True;
.     for i := 0 to High(groupmates) do
.       begin
.         if Length(groupmates[i]) >= StudentsAmountInt then
120           ListBox1.Items.Add('Группа ' + groupmates[i][0].Group +
.             '; Количество учащихся: ' + IntToStr(Length(groupmates[i])));
.         end;
.       end;

.   procedure TMainForm.Button6Click(Sender: TObject);
125   var i: integer;
.   begin
.     ListBox1.Clear;
.     ListBox1.Items.Add('Студенты, рейтинг которых неменьше 60');
130     for i := 0 to High(Repo.Students) do
.       if Repo.Students[i].Rating >= 60 then
.         ListBox1.Items.Add(Repo.Students[i].ToString);
.       end;

.   procedure TMainForm.Button7Click(Sender: TObject);
135   begin
.     GistForm.Show;
.   end;

```

Рисунок 19 - код модуля main

```

1  unit addStudent;
.
.  {$mode ObjFPC}{$H+}
.
5  interface
.
.  uses
.  [
.    Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls,
.    Domain, repository;
10 ]
.
.  type
.
.  [ { TAddStudentForm }
.
15 TAddStudentForm = class(TForm)
.    Button1: TButton;
.    Edit1: TEdit;
.    Edit2: TEdit;
.    Edit3: TEdit;
20    Edit4: TEdit;
.    procedure Button1Click(Sender: TObject);
.    procedure FormShow(Sender: TObject);
.  private
.
25 public
.
.    end;
.
29 var
.    |
30    AddStudentForm: TAddStudentForm;

```

Рисунок 20 - код модуля addStudent

```

implementation

{$R *.lfm}

[ { TAddStudentForm }

procedure TAddStudentForm.FormShow(Sender: TObject);
begin
    Edit1.Clear;
    Edit2.Clear;
    Edit3.Clear;
    Edit4.Clear;
end;

procedure TAddStudentForm.Button1Click(Sender: TObject);
var student: TStudent;
begin
    student.Init(edit1.text, edit2.text, edit3.text, StrToInt(edit4.text));
    Repo.Append(student);
    Hide;
end;

end.

```

Рисунок 21 - код модуля addStudent

```

1  unit gist;
.
.  {$mode delphi}{$H+}
.
5  interface
.
.  uses
.  [ Classes, SysUtils, Forms, Controls, Graphics, Dialogs, TAGraph, TASources,
.  [   TASeries, TACustomSource, TADbSource, Repository, Domain;
10
.  type
.
.  [ { TForm1 }
.
15 [ TForm1 = class(TForm)
.  [   Chart1: TChart;
.  [   Chart1BarSeries1: TBarSeries;
.  [   ChartSource: TRandomChartSource;
.  [   ListChartSource1: TListChartSource;
20 [   procedure FormShow(Sender: TObject);
.  [   function ListChartSource1Compare(AItem1, AItem2: Pointer): integer;
.  [ private
.  [
.  [ public
25 [
.  [   end;
.
.
.  var
.  [   GistForm: TForm1;
30
31 implementation
.  [ {$R *.lfm}
.
.  [ { TForm1 }
35
.  [ procedure TForm1.FormShow(Sender: TObject);
.  [ var groupmates: TGroupBy;
.  [   i, j: integer;
.  [   avr: double;
40 [ begin
.  [   groupmates := Repo.GroupBy(isGroupmates);
.  [   ListChartSource1.Clear;
.  [   for i := 0 to High(groupmates) do
.  [   [ begin
45 [     avr := 0;
.  [     for j := 0 to High(groupmates[i]) do
.  [       avr += groupmates[i][j].Rating;
.  [     avr /= Length(groupmates[i]);
.  [     ListChartSource1.Add(i, avr, groupmates[i][0].Group);
50 [   end;
.  [ end;
.

```

Рисунок 22 - код модуля gist

Выводы

В результате были получены практические навыки разработки прикладных приложений с графическим интерфейсом на языке программирования ObjectPascal с использованием библиотеки компонентов LCL.

Задание 2. Создание программной системы с элементарным интерфейсом консольного режима на C++

Задание

Выполнить структурную декомпозицию, разработать структурную схему, содержащую не менее 3 подпрограмм, и алгоритмы этих подпрограмм. Реализовать на C++ в консольном режиме. Предусмотреть примитивный интерфейс типа меню, позволяющий выбирать нужную подпрограмму.

Разработать программу, которая реализует операции над матрицами. Реализовать следующие операции: ввод матрицы, умножение матрицы на число, получение верхней треугольной и нижней треугольной матриц из заданной матрицы, а также вывод результатов операций на экран.

Цель

Получить практические навыки разработки cli-приложений на языке C++.

Проект программы

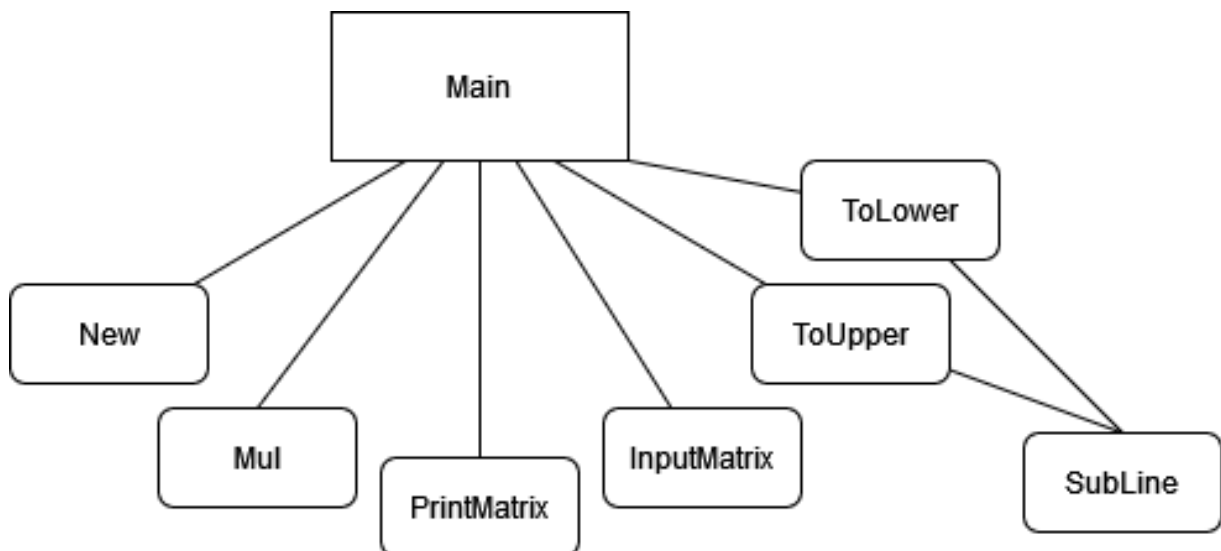


Рисунок 23 - структурная схема

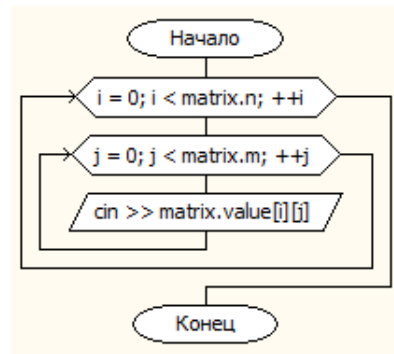


Рисунок 24 - схема алгоритма InputMatrix

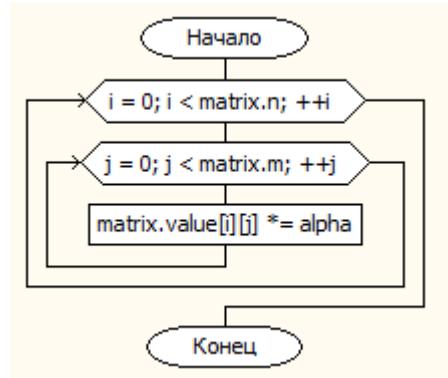


Рисунок 25 - схема алгоритма Mul

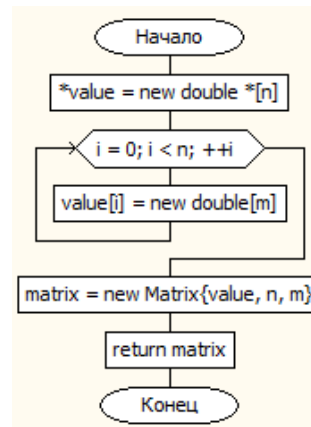


Рисунок 26 - схема алгоритма New

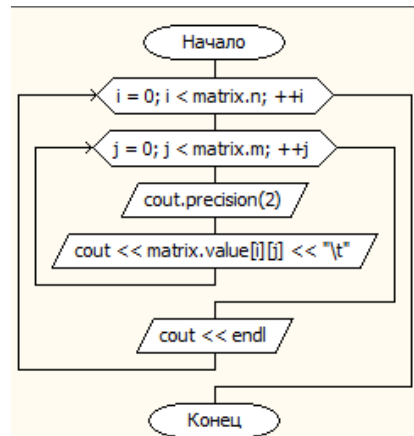


Рисунок 27 - схема алгоритма PrintMatrix

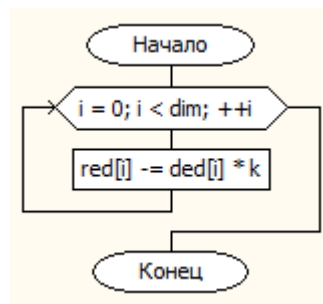


Рисунок 28 - схема алгоритма SubLine

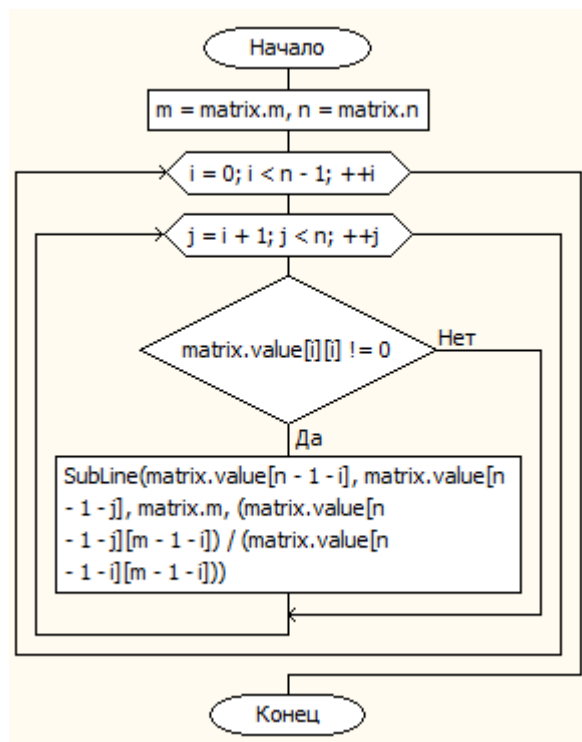


Рисунок 29 - схема алгоритма ToLower

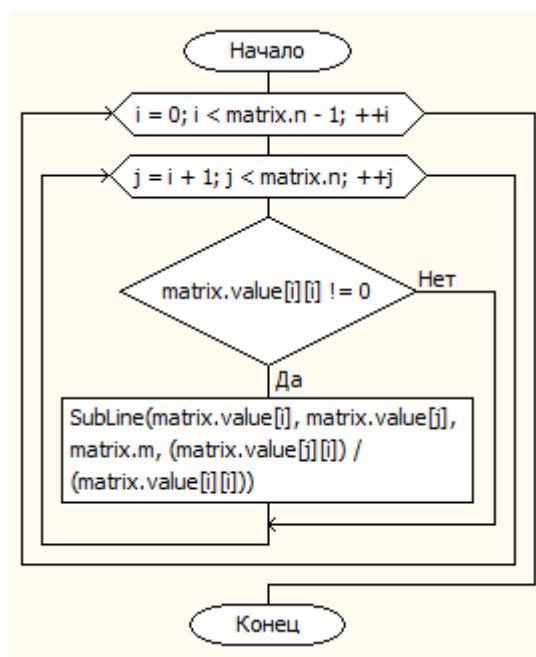


Рисунок 30 - схема алгоритма ToUpper

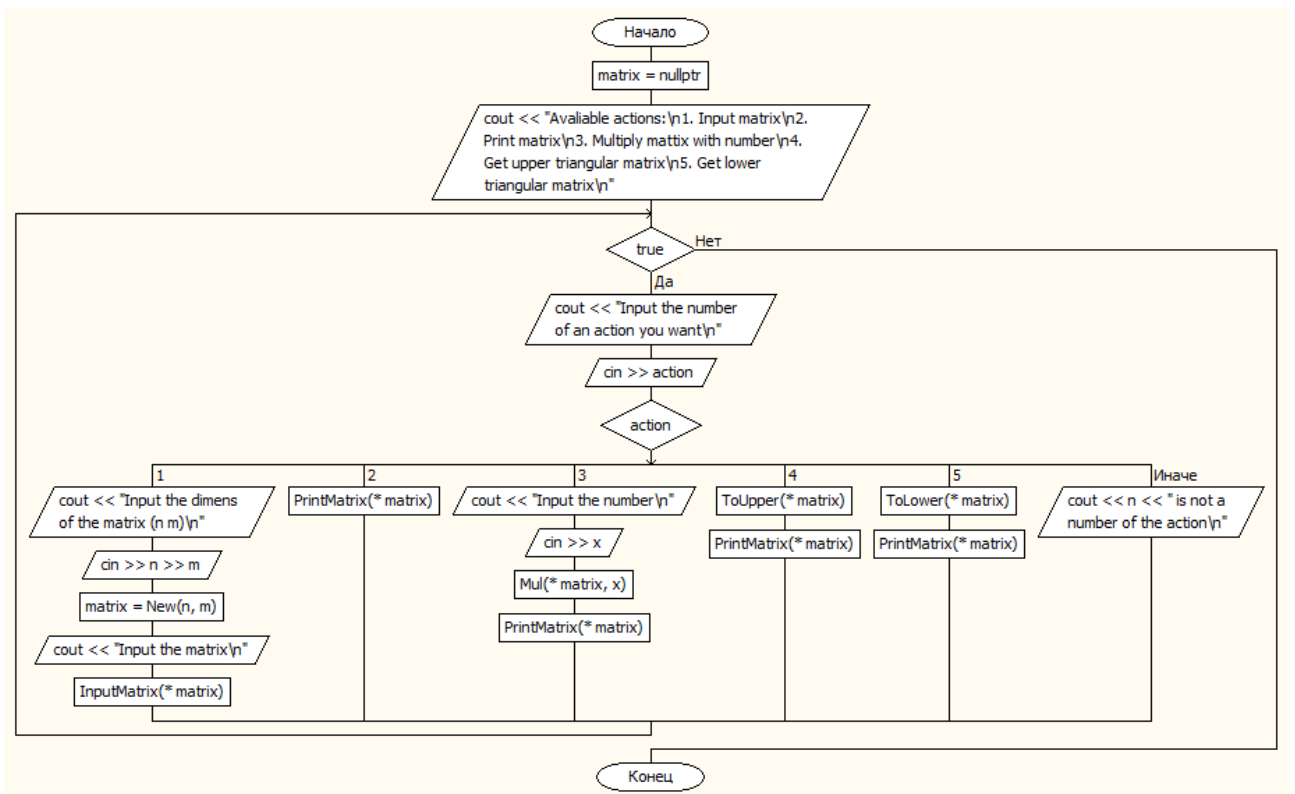


Рисунок 31 - схема алгоритма Main

Код программы

```

1  #include <iostream>
2
3  using namespace std;
4
5  typedef struct Matrix {
6      double** value;
7      int n;
8      int m;
9  } Matrix;
10
11  Matrix* New(int n, int m) {
12      double** value = new double*[n];
13      for (size_t i = 0; i < n; ++i) {
14          value[i] = new double[m];
15      }
16      Matrix* matrix = new Matrix{
17          value, n, m
18      };
19      return matrix;
20  }
21
22  void Mul(Matrix& matrix, double alpha) {
23      for (size_t i = 0; i < matrix.n; ++i) {
24          for (size_t j = 0; j < matrix.m; ++j) {
25              matrix.value[i][j] *= alpha;
26          }
27      }
28  }

```

Рисунок 32 - код программы

```

30 void PrintMatrix(Matrix& matrix) {
31     for (size_t i = 0; i < matrix.n; ++i) {
32         for (size_t j = 0; j < matrix.m; ++j) {
33             cout.precision(2);
34             cout << matrix.value[i][j] << "\t";
35         }
36         cout << endl;
37     }
38 }
39
40 void InputMatrix(Matrix& matrix) {
41     for (size_t i = 0; i < matrix.n; ++i) {
42         for (size_t j = 0; j < matrix.m; ++j) {
43             cin >> matrix.value[i][j];
44         }
45     }
46 }
47
48 void SubLine(double*& ded, double*& red, int dim, double k) {
49     for (size_t i = 0; i < dim; ++i) {
50         red[i] -= ded[i] * k;
51     }
52 }
53
54 void ToUpper(Matrix& matrix) {
55     for (size_t i = 0; i < matrix.n-1; ++i) {
56         for (size_t j = i+1; j < matrix.n; ++j) {
57             if (matrix.value[i][i] != 0) {
58                 SubLine(matrix.value[i],
59                     matrix.value[j],
60                     matrix.m,
61                     (matrix.value[j][i])/(matrix.value[i][i])
62                 );
63             }
64         }
65     }
66 }

```

Рисунок 33 - код программы

```

68 void ToLower(Matrix& matrix) {
69     int m = matrix.m, n = matrix.n;
70     for (size_t i = 0; i < n-1; ++i) {
71         for (size_t j = i+1; j < n; ++j) {
72             if (matrix.value[i][i] != 0) {
73                 SubLine(matrix.value[n-1-i],
74                     matrix.value[n-1-j],
75                     matrix.m,
76                     (matrix.value[n-1-j][m-1-i])/(matrix.value[n-1-i][m-1-i])
77                 );
78             }
79         }
80     }
81 }

```

Рисунок 34 - код программы

```

83 int main() {
84     Matrix* matrix = nullptr;
85     int n, m;
86
87     cout << "Avaliable actions:\n1. Input matrix\n2. Print matrix\n3." +
88         "\n4. Multiply mattix with number\n5. Get upper triangular matrix\n5. Get lower triangular matrix\n";
89
90     while(true) {
91         cout << "Input the number of an action you want\n";
92         int action;
93         cin >> action;
94         switch(action) {
95             case 1:
96                 cout << "Input the dimens of the matrix (n m)\n";
97                 cin >> n >> m;
98                 matrix = New(n, m);
99                 cout << "Input the matrix\n";
100                 InputMatrix(*matrix);
101                 break;
102             case 2:
103                 PrintMatrix(*matrix);
104                 break;
105             case 3:
106                 double x;
107                 cout << "Input the number\n";
108                 cin >> x;
109                 Mul(*matrix, x);
110                 PrintMatrix(*matrix);
111                 break;
112             case 4:
113                 ToUpper(*matrix);
114                 PrintMatrix(*matrix);
115                 break;
116             case 5:
117                 ToLower(*matrix);
118                 PrintMatrix(*matrix);
119                 break;
120             default:
121                 cout << n << " is not a number of the action\n";
122         }
123     }
124 }

```

Рисунок 35 - код программы

Тестирование программы

```

> ./main
Avaliable actions:
1. Input matrix
2. Print matrix
3. Multiply mattix with number
4. Get upper triangular matrix
5. Get lower triangular matrix
Input the number of an action you want
1
Input the dimens of the matrix (n m)
2 3
Input the matrix
1 2 3
4 5 6
Input the number of an action you want
2
1 2 3
4 5 6
Input the number of an action you want
3
Input the number
4
4 8 12
16 20 24
Input the number of an action you want
4
4 8 12
0 -12 -24

```

Рисунок 36 - пример работы программы

Вывод

Были получены практические навыки разработки cli-приложений на языке C++.

Задание 3. Создание программной системы с Qt интерфейсом на C++

Выполнить объектную декомпозицию, разработать формы интерфейса, диаграмму состояний интерфейса, диаграммы классов интерфейсной и предметной областей, диаграмму последовательности действий одной из реализуемых операций. Разработать, протестировать и отладить программу в среде Visual Studio или QT Creator.

Сведения о студентах включают: фамилию, имя, индекс группы, рейтинг (от 0 до 100). Программа должна в интерактивном режиме формировать файл, добавлять и удалять данные, а также воспринимать каждый из перечисленных запросов и давать на него ответ.

1. Выяснить, имеются ли в институте однофамильцы. Если имеются, показать их фамилии и имена.
2. Выяснить, в каких группах обучается количество студентов более заданного.
3. Получить список студентов с указанием группы, рейтинг которых не меньше зачетного (60).
4. Построить гистограмму, показывающую средний рейтинг по каждой группе.

Цель

Разработать настольное приложение на языке C++ с использованием библиотеки Qt. Изучить основы работы с библиотекой графических интерфейсов Qt, а также средой разработки Qt Creator.

Проект программы

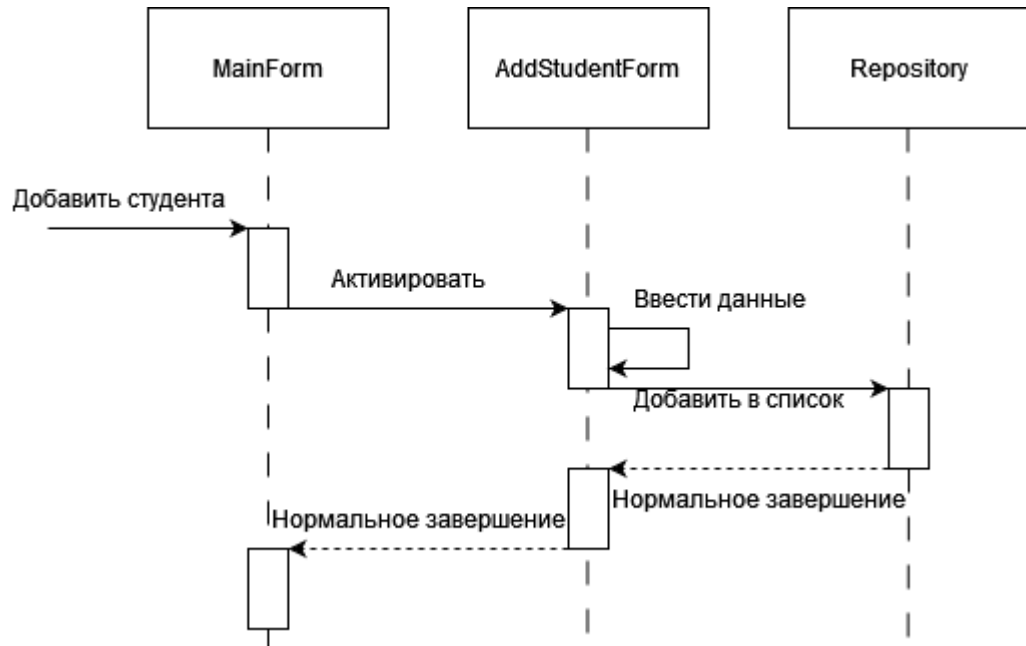


Рисунок 37 - диаграмма последовательности действий при добавлении нового пользователя

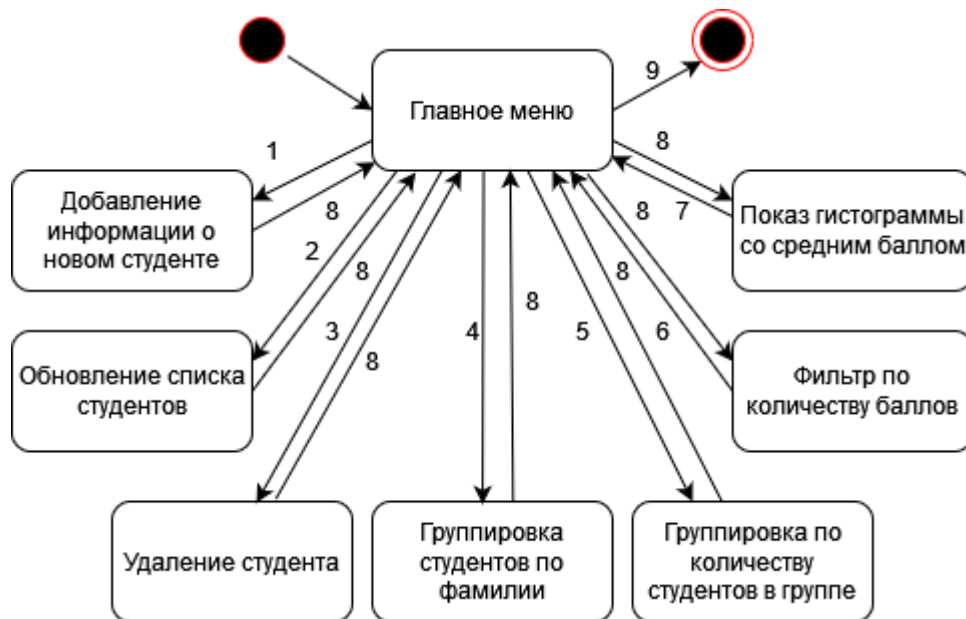


Рисунок 38 - диаграмма состояний

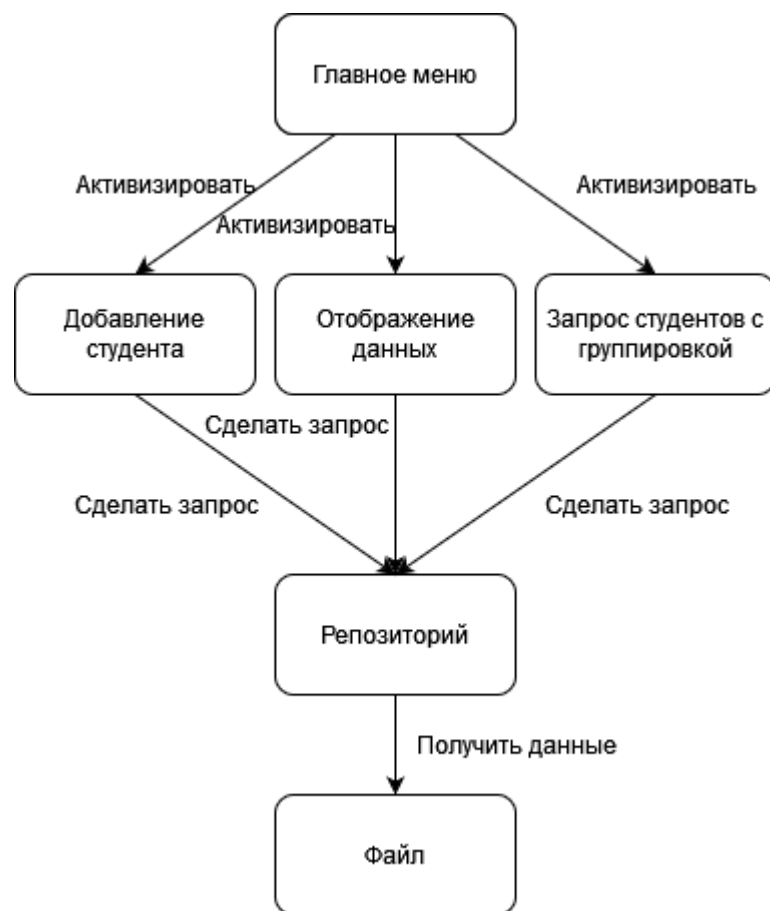


Рисунок 39 - объектная декомпозиция

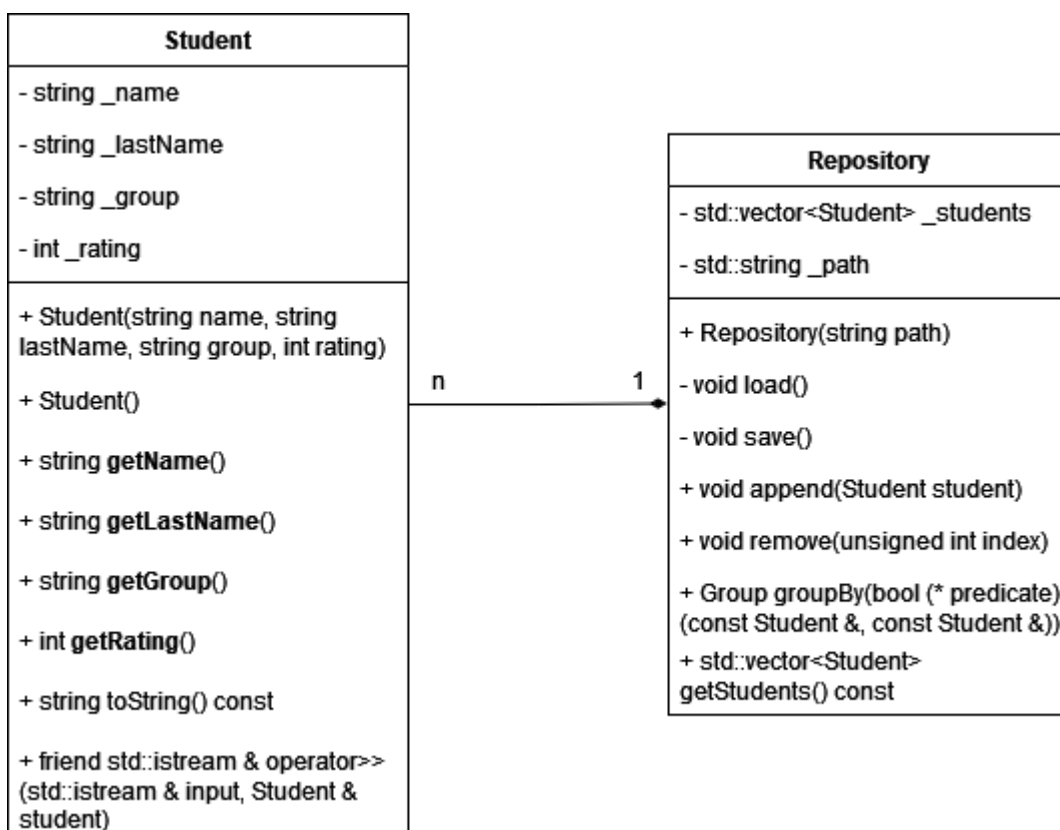


Рисунок 40 - диаграмма классов предметной области

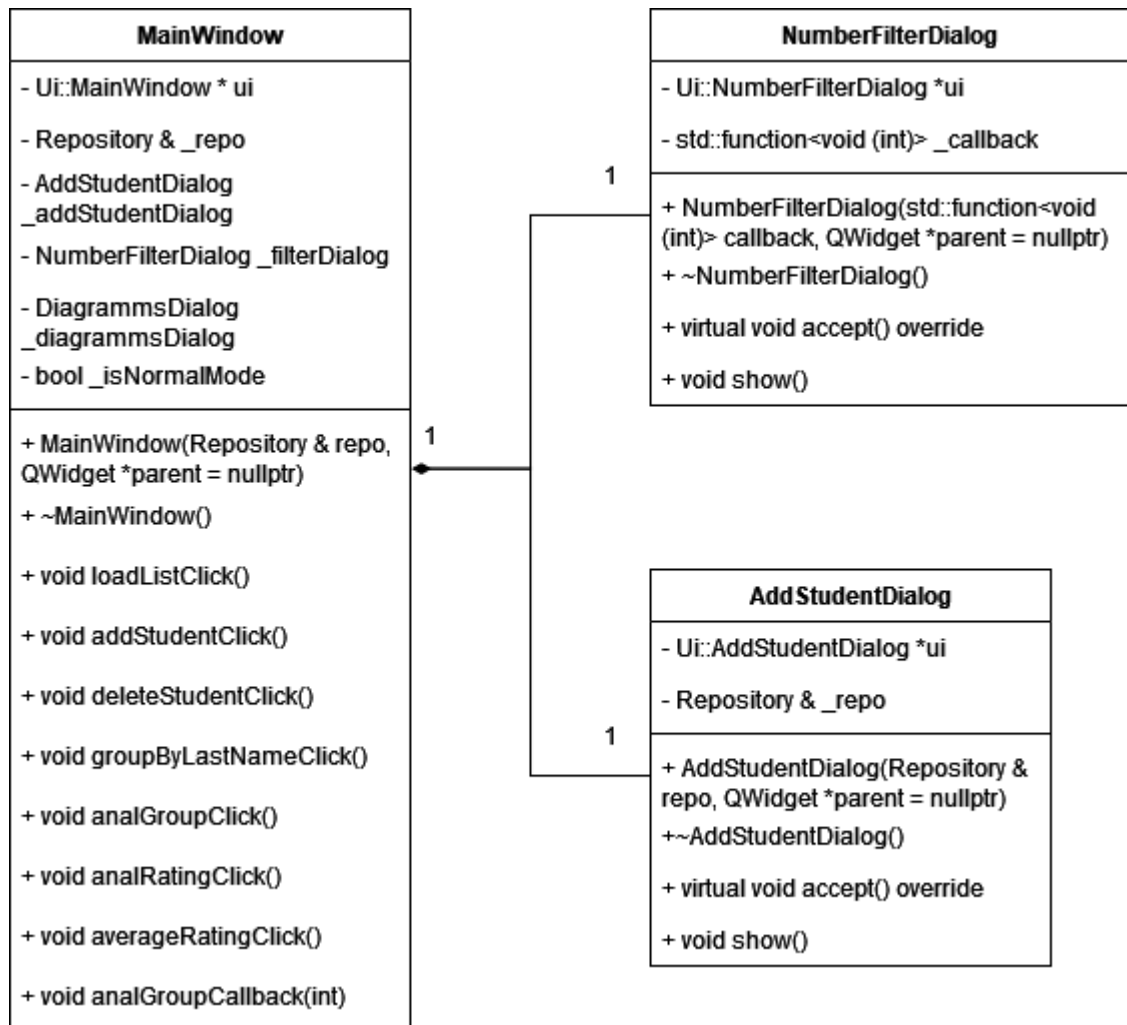


Рисунок 41 - диаграмма классов интерфейса

Интерфейс приложения

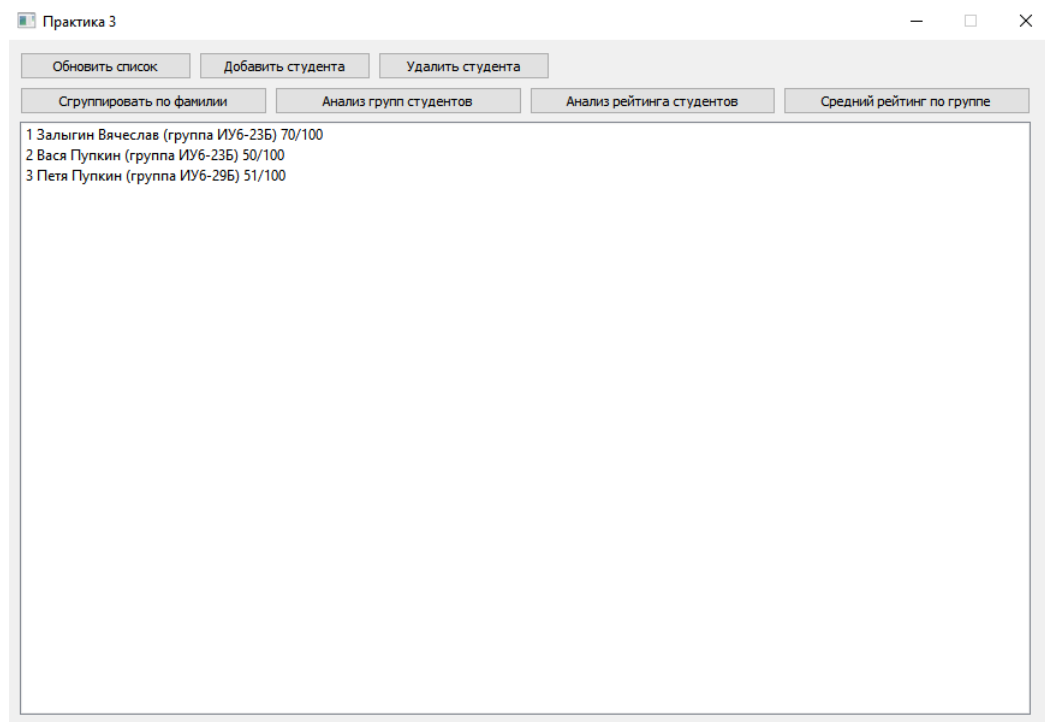


Рисунок 42 - главное меню

Dialog

Имя
Новый

Фамилия
Студент

Группа
Бауманки

Рейтинг (от 0 до 100)
0

OK Cancel

Рисунок 43 - диалог добавления нового студента

Практика 3

Обновить список Добавить студента Удалить студента

Сгруппировать по фамилии Анализ групп студентов Анализ рейтинга студентов Средний рейтинг по группе

Фамилия: Пупкин
Вася Пупкин (группа ИУ6-23Б) 50/100
Петя Пупкин (группа ИУ6-29Б) 51/100

Фамилия: Залыгин
Вячеслав Залыгин (группа ИУ6-23Б) 70/100

Рисунок 44 - режим группировки по фамилиям

Практика 3

Обновить список Добавить студента Удалить студента

Сгруппировать по фамилии Анализ групп студентов Анализ рейтинга студентов Средний рейтинг по группе

Группы, где человек больше 1
Группа: ИУ6-23Б
Вася Пупкин (группа ИУ6-23Б) 50/100
Вячеслав Залыгин (группа ИУ6-23Б) 70/100

Рисунок 45 - режим фильтрации групп по количеству человек

Практика 3

Обновить список Добавить студента Удалить студента

Сгруппировать по фамилии Анализ групп студентов Анализ рейтинга студентов Средний рейтинг по группе

Студенты, у которых меньше 60 баллов:
Вася Пупкин (группа ИУ6-23Б) 50/100
Петя Пупкин (группа ИУ6-29Б) 51/100

Рисунок 46 - режим фильтрации по рейтингу студента



Рисунок 47 - диалог с диаграммой средних по группе баллов

Код программы

```
addstudentdialog.hpp
1  #ifndef ADDSTUDENTDIALOG_H
2  #define ADDSTUDENTDIALOG_H
3
4  #include <QDialog>
5  #include <sstream>
6  #include <memory>
7  #include "repository.hpp"
8
9  namespace Ui {
10     class AddStudentDialog;
11 }
12
13 class AddStudentDialog : public QDialog
14 {
15     Q_OBJECT
16
17 public:
18     explicit AddStudentDialog(Repository & repo, QWidget *parent = nullptr);
19     ~AddStudentDialog();
20
21     virtual void accept() override;
22     void show();
23
24 private:
25     Ui::AddStudentDialog *ui;
26     Repository & _repo;
27 };
28
29 #endif // ADDSTUDENTDIALOG_H
```

Рисунок 48

```

1  #ifndef DIAGRAMMSDIALOG_H
2  #define DIAGRAMMSDIALOG_H
3
4  #include "repository.hpp"
5
6  #include <QDialog>
7  #include <QBarSet>
8  #include <QChart>
9  #include <QBarSeries>
10 #include <QValueAxis>
11 #include <QChartView>
12
13 using namespace QtCharts;
14
15 namespace Ui {
16 class DiagrammsDialog;
17 }
18
19 class DiagrammsDialog : public QDialog
20 {
21     Q_OBJECT
22
23 public:
24     DiagrammsDialog(Repository & repo, QWidget *parent = nullptr);
25     ~DiagrammsDialog();
26
27     void show();
28
29 private:
30     QChartView * chartView;
31     Repository & _repo;
32 };
33
34 #endif // DIAGRAMMSDIALOG_H
35

```

Рисунок 49

```

10 #include <QMainWindow>
11
12 QT_BEGIN_NAMESPACE
13 namespace Ui { class MainWindow; }
14 QT_END_NAMESPACE
15
16 class MainWindow : public QMainWindow
17 {
18     Q_OBJECT
19
20 public:
21     MainWindow(Repository & repo, QWidget *parent = nullptr);
22     ~MainWindow() {
23         delete ui;
24     };
25
26 public slots:
27     void loadListClick();
28     void addStudentClick();
29     void deleteStudentClick();
30     void groupByLastNameClick();
31     void analGroupClick();
32     void analRatingClick();
33     void averageRatingClick();
34     void analGroupCallback(int);
35
36 private:
37     Ui::MainWindow * ui;
38     Repository & _repo;
39     AddStudentDialog _addStudentDialog;
40     NumberFilterDialog _filterDialog;
41     DiagrammsDialog _diagrammsDialog;
42     bool _isNormalMode;
43 };
44 #endif // MAINWINDOW_H
45

```

Рисунок 50

```

> numberfilterdialog.hpp <No Symbols>
1  #ifndef NUMBERFILTERDIALOG_H
2  #define NUMBERFILTERDIALOG_H
3
4  #include <QDialog>
5  #include <functional>
6
7  namespace Ui {
8      class NumberFilterDialog;
9  }
10
11  class NumberFilterDialog : public QDialog
12  {
13      Q_OBJECT
14
15  public:
16      explicit NumberFilterDialog(std::function<void (int)> callback, QWidget *parent = nullptr)
17          : NumberFilterDialog();
18
19      virtual void accept() override;
20      void show();
21  private:
22      Ui::NumberFilterDialog *ui;
23
24      std::function<void (int)> _callback;
25  };
26
27  #endif // NUMBERFILTERDIALOG_H

```

Рисунок 51

```

> repository.hpp Repository
1  #ifndef REPOSITORY_H
2  #define REPOSITORY_H
3
4  #include <vector>
5  #include <string>
6
7  #include <student.hpp>
8
9
10 typedef std::vector<std::vector<Student>> Group;
11
12 class Repository
13 {
14 private:
15     std::vector<Student> _students;
16     std::string _path;
17
18     void load();
19     void save();
20
21 public:
22     Repository(string path)
23         : _path(path) { load(); };
24
25     void append(Student student);
26     void remove(unsigned int index);
27     Group groupBy(bool (* predicate)(const Student &, const Student &));
28     std::vector<Student> getStudents() const { return _students; }
29 };
30 #endif // REPOSITORY_H

```

Рисунок 52

```

> student.hpp
1  #ifndef STUDENT_H
2  #define STUDENT_H
3
4  #include <string>
5  #include <fstream>
6
7  using std::string;
8
9  class Student
10 {
11     private:
12         string _name;
13         string _lastName;
14         string _group;
15         int _rating;
16     public:
17         Student(string name, string lastName, string group, int rating)
18             : _name(name), _lastName(lastName), _group(group), _rating(rating) {};
19
20         Student()
21             : _name("empty"), _lastName("empty"), _group("empty"), _rating(-1) {};
22
23         string getName() const { return _name; }
24         string getLastName() const { return _lastName; }
25         string getGroup() const { return _group; }
26         int getRating() const { return _rating; }
27
28         string toString() const;
29
30         friend std::istream & operator>>(std::istream & input, Student & student);
31     };
32
33     bool operator==(Student & left, Student & right);
34
35     std::istream & operator>>(std::istream & input, Student & student);
36     std::ostream & operator<<(std::ostream & output, const Student & student);
37
38 #endif // STUDENT_H

```

Рисунок 53


```

1  #include "addstudentdialog.hpp"
2  #include "ui_addstudentdialog.h"
3
4  #include <QMessageBox>
5
6  AddStudentDialog::AddStudentDialog(Repository & repo, QWidget *parent)
7      : QDialog(parent), ui(new Ui::AddStudentDialog), _repo(repo)
8  {
9      ui->setupUi(this);
10 }
11
12 AddStudentDialog::~AddStudentDialog()
13 {
14     delete ui;
15 }
16
17 bool checkStr(std::string str) {
18     for (const auto & e : str) {
19         if (e == ' ') {
20             return false;
21         }
22     }
23     return true;
24 }
25
26 bool checkRating(std::string rating) {
27     std::stringstream ss(rating);
28     int a;
29     if ((ss >> a).fail() || !(ss >> std::ws).eof())
30         return false;
31     return a >= 0 && a <= 100;
32 }
33
34 void AddStudentDialog::accept() {
35     if (
36         checkStr(ui->nameEdit->text().toStdString()) &&
37         checkStr(ui->lastNameEdit->text().toStdString()) &&
38         checkStr(ui->groupEdit->text().toStdString()) &&
39         checkRating(ui->ratingEdit->text().toStdString()) {
40         int rating;
41         std::stringstream ss(ui->ratingEdit->text().toStdString());
42         ss >> rating;
43         Student s(ui->nameEdit->text().toStdString(), ui->lastNameEdit->text().toStdString(), ui->groupEdit->text().toStdString(), rating);
44         _repo.append(std::move(s));
45         QDialog::accept();
46     } else {
47         QMessageBox msg(QMessageBox::Information, "Некорректные данные", "Некорректный формат данных.", QMessageBox::Ok);
48         msg.exec();
49     }
50 }
51
52 void AddStudentDialog::show()
53 {
54     ui->nameEdit->clear();
55     ui->lastNameEdit->clear();
56     ui->groupEdit->clear();
57     ui->ratingEdit->clear();
58     QDialog::show();
59 }

```

Рисунок 54

```

1  #include "diagrammsdialog.hpp"
2
3  DiagrammsDialog::DiagrammsDialog(Repository & repo, QWidget *parent) :
4      QDialog(parent), _repo(repo)
5  {
6      resize(400, 400);
7  }
8
9  DiagrammsDialog::~DiagrammsDialog() { }
10
11 bool sameGroup(const Student & lhs, const Student & rhs) { return lhs.getGroup() == rhs.getGroup(); }
12
13 void DiagrammsDialog::show()
14 {
15     QBarSeries * series = new QBarSeries(this);
16     for (const auto & group : _repo.groupBy(sameGroup)) {
17         int average = 0;
18         for (const auto & student : group) { average += student.getRating(); }
19         average /= group.size();
20         average += 1;
21         QBarSet * bar = new QBarSet(QString::fromStdString(group[0].getGroup()));
22         *bar << average;
23         series->append(bar);
24     }
25     QChart * chart = new QChart();
26     chart->addSeries(series);
27     chart->setTitle("Средний рейтинг по группам");
28     chart->setAnimationOptions(QChart::SeriesAnimations);
29
30     QValueAxis *axisY = new QValueAxis();
31     axisY->setRange(0,100);
32     chart->addAxis(axisY, Qt::AlignLeft);
33     series->attachAxis(axisY);
34
35     chart->legend()->setVisible(true);
36     chart->legend()->setAlignment(Qt::AlignBottom);
37
38     chartView = new QChartView(chart, this);
39     chartView->setRenderHint(QPainter::Antialiasing);
40     chartView->resize(400, 400);
41     QDialog::show();
42 }
43

```

Рисунок 55

```

1  #include "mainwindow.hpp"
2
3  #include <QApplication>
4
5  #include <repository.hpp>
6
7  int main(int argc, char *argv[])
8  {
9      Repository repo("./data.dat");
10     QApplication a(argc, argv);
11     MainWindow w(repo);
12     w.show();
13     return a.exec();
14 }
15

```

Рисунок 56

```

mainwindow.cpp
1 #include "mainwindow.hpp"
2 #include "ui_mainwindow.h"
3 #include "numberfilterdialog.hpp"
4
5 #include <QMessageBox>
6 #include <QString>
7 #include <vector>
8 #include <sstream>
9 #include <string>
10
11 MainWindow::MainWindow(Repository & repo, QWidget *parent)
12 : QMainWindow(parent), ui(new Ui::MainWindow), _repo(repo), _addStudentDialog(_repo, this), _isNormalMode(false),
13 _filterDialog([this](int filter) { this->analGroupCallback(filter); }, this), _diagramsDialog(_repo, this)
14 {
15     ui->setupUi(this);
16     connect(ui->addStudentButton, SIGNAL(clicked(bool)), this, SLOT(addStudentClick()));
17     connect(ui->loadListButton, SIGNAL(clicked(bool)), this, SLOT(loadListClick()));
18     connect(ui->deleteStudentButton, SIGNAL(clicked(bool)), this, SLOT(deleteStudentClick()));
19     connect(ui->groupByLastNameButton, SIGNAL(clicked(bool)), this, SLOT(groupByLastNameClick()));
20     connect(ui->analGroupButton, SIGNAL(clicked(bool)), this, SLOT(analGroupClick()));
21     connect(ui->analRatingButton, SIGNAL(clicked(bool)), this, SLOT(analRatingClick()));
22     connect(ui->averageRatingButton, SIGNAL(clicked(bool)), this, SLOT(averageRatingClick()));
23 }
24
25 void MainWindow::loadListClick() {
26     std::vector<Student> students = _repo.getStudents();
27     ui->list->clear();
28     for (unsigned int i = 0; i < students.size(); ++i) {
29         ui->list->addItem(QString::fromStdString(std::to_string(i+1) + " " + students[i].toString()));
30     }
31     _isNormalMode = true;
32 }
33
34 void MainWindow::addStudentClick() {
35     _addStudentDialog.show();
36 }
37
38 void MainWindow::deleteStudentClick() {
39     if (_isNormalMode) {
40         std::stringstream ss(ui->list->currentItem()->text().toStdString());
41         int i;
42         ss >> i;
43         _repo.remove(i-1);
44         loadListClick();
45     }
46 }
47
48 bool equalsByLastName(const Student & lhs, const Student & rhs) { return lhs.getLastName() == rhs.getLastName(); }
49

```

Рисунок 57

```

numberfilterdialog.cpp
1 #include "numberfilterdialog.hpp"
2 #include "ui_numberfilterdialog.h"
3
4 #include <QMessageBox>
5
6 #include <sstream>
7
8 NumberFilterDialog::NumberFilterDialog(std::function<void (int)> callback, QWidget *parent) :
9 QDialog(parent), ui(new Ui::NumberFilterDialog), _callback(callback)
10 {
11     ui->setupUi(this);
12 }
13
14 NumberFilterDialog::~NumberFilterDialog()
15 {
16     delete ui;
17 }
18
19 bool parseFilter(std::string filter, int & result) {
20     std::stringstream ss(filter);
21     int a;
22     if ((ss >> a).fail() || !(ss >> std::ws).eof())
23         return false;
24     result = a;
25     return a >= 0;
26 }
27
28 void NumberFilterDialog::accept()
29 {
30     int filter;
31     if (parseFilter(ui->filterEdit->text().toStdString(), filter)) {
32         _callback(filter);
33         QDialog::accept();
34     } else {
35         QMessageBox msg(QMessageBox::Information, "Некорректные данные", "Некорректный формат данных.", QMessageBox::Ok);
36         msg.exec();
37     }
38 }
39
40 void NumberFilterDialog::show() {
41     ui->filterEdit->clear();
42     QDialog::show();
43 }
44

```

Рисунок 58

```

> repository.cpp Repository::load() -> void
1  #include "repository.hpp"
2
3  #include <fstream>
4  void Repository::load() {
5      Student s;
6      _students.clear();
7
8      std::ifstream in(_path, std::ios::in);
9      while (in >> s) {
10         _students.push_back(s);
11     }
12     in.close();
13 }
14
15 void Repository::save() {
16     std::ofstream out(_path, std::ios::out | std::ios::trunc);
17     for (const auto & student : _students) {
18         out << student;
19     }
20     out.close();
21 }
22
23 void Repository::append(Student student) {
24     _students.push_back(student);
25     save();
26 }
27
28 void Repository::remove(unsigned int index) {
29     _students.erase(_students.begin()+index);
30     save();
31 }
32
33 Group Repository::groupBy(bool (*predicate)(const Student &, const Student &)) {
34     Group res;
35
36     for (const auto & student : _students) {
37         bool isInGroup = false;
38         for (auto & group : res) {
39             if (predicate(student, group[0]) && !isInGroup) {
40                 group.push_back(Student(student));
41                 isInGroup = true;
42             }
43         }
44         if (!isInGroup) {
45             res.push_back({Student(student)});
46         }
47     }
48
49     return res;
50 }

```

Рисунок 59

```

> student.cpp <Select Symbol>
1  #include "student.hpp"
2
3  bool operator==(Student & left, Student & right) {
4      return (left.getName() == right.getName()) &&
5             (left.getLastName() == right.getLastName()) &&
6             (left.getGroup() == right.getGroup()) &&
7             (left.getRating() == right.getRating());
8  }
9
10 string Student::toString() const {
11     return _name + " " + _lastName + " (rpyrna " + _group + ") " + std::to_string(_rating) + "/100";
12 }
13
14 std::istream & operator>>(std::istream & input, Student & student) {
15     input >> student._name >> student._lastName >> student._group >> student._rating;
16     return input;
17 }
18
19 std::ostream & operator<<(std::ostream & output, const Student & student) {
20     output << student.getName() << " " << student.getLastName() << " " << student.getGroup() << " " << student.getRating() << " ";
21     return output;
22 }
23

```

Рисунок 60

Вывод

Было разработано настольное приложение на языке C++ с использованием библиотеки Qt. Были изучены основы работы с библиотекой графических интерфейсов Qt, а также средой разработки Qt Creator.