

Содержание

| | |
|--|----|
| Введение..... | 4 |
| 1. Общие сведения..... | 5 |
| 1.1 Обозначение и наименование программы | 5 |
| 1.2 Программное обеспечение, необходимое для функционирования программы..... | 5 |
| 1.3 Языки программирования, на которых написана программа | 5 |
| 2. Функциональное назначение | 5 |
| 3. Описание логической структуры..... | 6 |
| 3.1 Декомпозиция | 6 |
| 3.2 Описание зависимостей | 7 |
| 3.3 Описание деталей | 7 |
| 3.3.1 Класс Widget..... | 7 |
| 3.3.2 Класс Input | 14 |
| 3.3.3 Класс Uni..... | 15 |
| 3.4 Описание графического интерфейса | 16 |
| Заключение | 21 |
| Список литературы | 21 |
| Приложение 1. Техническое задание | 22 |

Введение

В качестве цели курсовой работы была выбрана разработка приложения для поиска направлений подготовки в университетах Москвы на основе баллов ЕГЭ. Данная тема была выбрана из-за необходимости создания приложения, которое помогло бы абитуриенту выбрать образовательную программу.

При разработке курсовой работы планируется освоить профессиональные компетенции ПК-13.

1. Общие сведения

В контексте данной курсовой работы было разработано приложение для поиска направлений подготовки в университетах Москвы на основе баллов ЕГЭ.

1.1 Обозначение и наименование программы

Наименование программного приложения — “USE Calculator”.

1.2 Программное обеспечение, необходимое для функционирования программы

Для функционирования приложения необходимо установить Qt Creator.

1.3 Языки программирования, на которых написана программа

Данная программа написана на языке программирования C++.

C++ – язык программирования общего назначения, который поддерживает такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование. C++ сочетает свойства как высокоуровневых, так и низкоуровневых языков.

Qt – кроссплатформенный фреймворк для разработки программного обеспечения на языке программирования C++. Включает в себя все основные классы, которые могут потребоваться при разработке прикладного программного обеспечения, начиная от элементов графического интерфейса и заканчивая классами для работы с сетью, базами данных и XML.

2. Функциональное назначение

Приложение «USE Calculator» предназначено для поиска образовательных программ уровня бакалавриата и специалитета в вузах Москвы. Приложению требуется предоставлять пользователю возможность просматривать информацию, удовлетворяющую условиям поиска, в виде таблицы. Также требуется обеспечивать пользователю возможность добавлять новую информацию о направлениях подготовки.

3. Описание логической структуры

3.1 Декомпозиция

Исходный код приложения состоит из нескольких классов, иерархия которых изображена на рисунке 1.

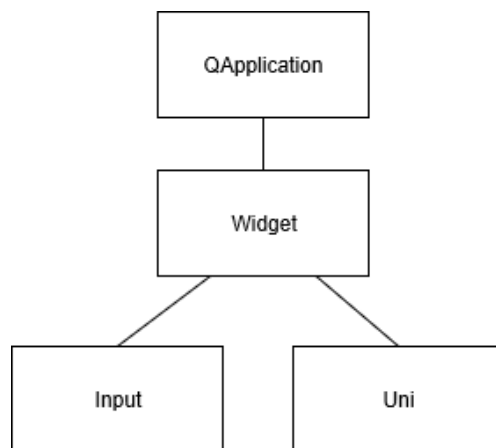


Рис.1 Система классов приложения

QApplication – класс, организующий цикл сбора сообщений, которые будут приходить в окно нашего приложения.

Widget – класс, отвечающий за визуальное и программное оформление приложения. В методах этого класса описывается логика работы кнопок, полей для ввода и прочего, также производится настройка интерфейса.

Input – класс, объект которого содержит в себе информацию, введенную пользователем через форму.

Uni – класс, предназначенный для работы с записями файла, в котором хранится информация об образовательных программах. Полностью состоит из полей, содержимым которых заполняется результирующая таблица.

widget.ui – UI-файл Qt Designer, представляющий собой дерево виджетов формы в формате XML.

ege.txt – текстовый файл, содержащий информацию о направлениях подготовки в московских университетах. Информацией из этого файла заполняется результирующая таблица.

3.2 Описание зависимостей

В контексте разработки приложения был использован фреймворк Qt, с помощью которого были созданы все элементы графического интерфейса.

3.3 Описание деталей

3.3.1 Класс Widget

На рисунке 2 представлена UML-диаграмма класса Widget, производного от класса QWidget. Класс QWidget предоставляется фреймворком Qt. Этот класс является простейшим виджетом. В терминологии Qt это простейший элемент, с которым связана какая-то графическая область на экране. Используется как базовый класс для главного окна.

Класс Widget создан для настройки внешнего вида окна приложения и для работы с поступающими от пользователя данными.

Класс содержит пять методов: конструктор Widget, деструктор ~Widget, метод-обработчик нажатия кнопки поиска on_ButtonSearch_released, метод-обработчик нажатия кнопки добавления новой информации on_AddButton_released, метод-обработчик нажатия кнопки сохранения введенной пользователем информации on_SaveButton_released.

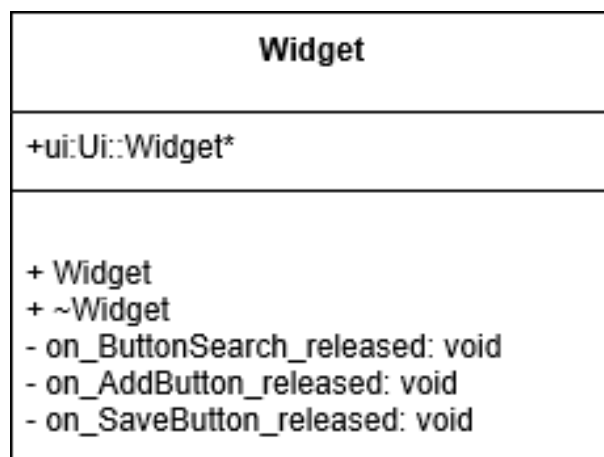


Рис. 2 UML-диаграмма класса Widget

3.3.1.1 Метод Widget

Данный метод устанавливает пользовательский интерфейс в соответствии с описанием в ui файле (см. п.3.1.), устанавливает размеры таблицы для вывода информации, удовлетворяющей запросу пользователя, заголовки столбцов, определяет внешний вид окна при запуске, т.е. скрывает или, наоборот, делает видимыми некоторые элементы интерфейса.

3.3.1.2 Метод ~Widget

~Widget – деструктор класса Widget. Разрушает объект данного класса.

3.3.1.3 Метод on_ButtonSearch_released

Данный метод обрабатывает нажатие кнопки “Search”.

Ниже, на рисунке 3, представлена блок-схема алгоритма, который выполняется при нажатии вышеупомянутой кнопки.

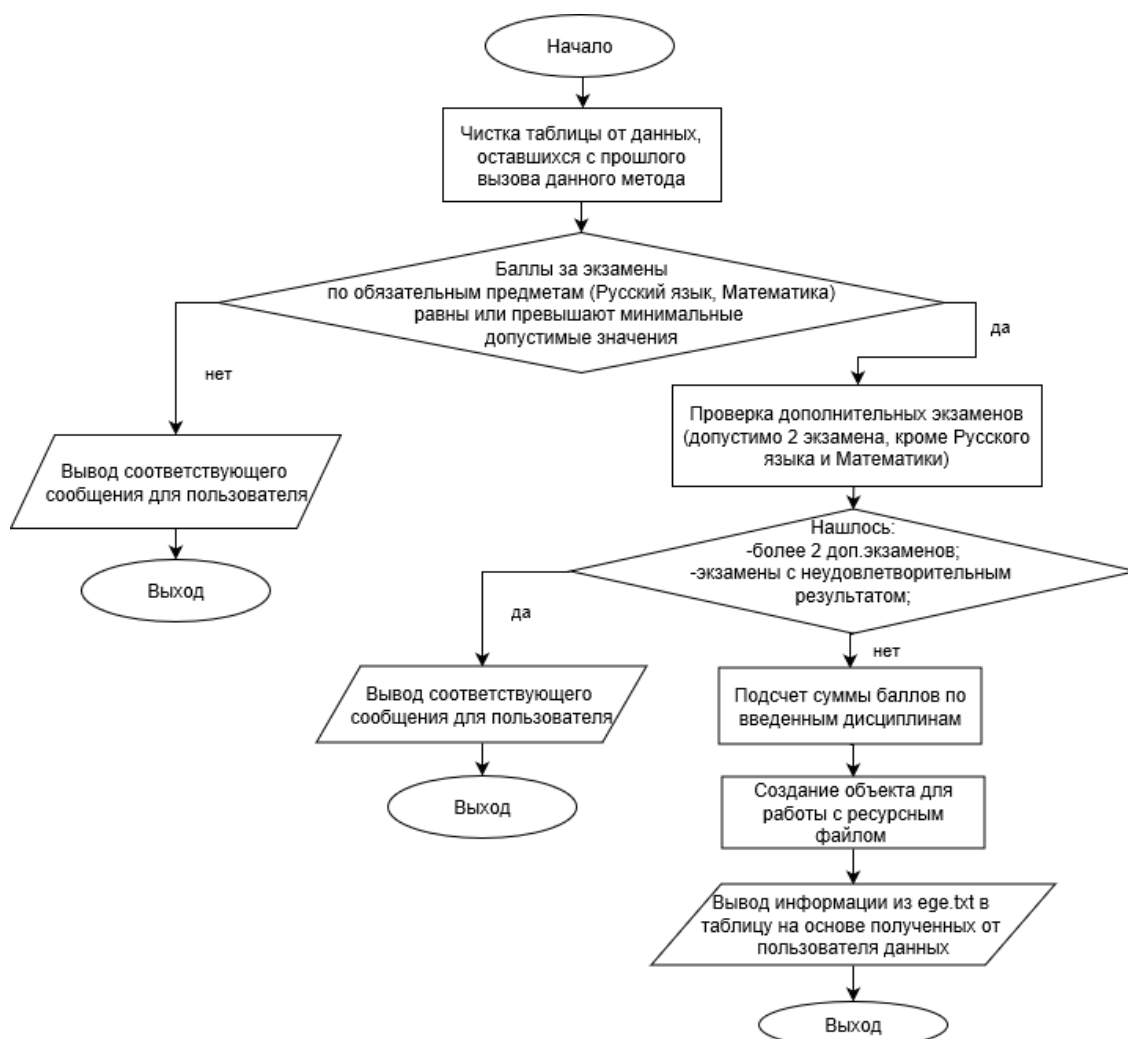


Рис. 3 Блок-схема метода on_ButtonSearch_released

Рассмотрим подробнее некоторые этапы работы данного метода.

Проверка дополнительных экзаменов:

Идея состоит в том, чтоб в цикле проверить содержимое всех полей для ввода баллов; затем, в зависимости от того, каким по порядку является текущее поле, определить минимальный балл для него.

```
//Экзамены по выбору:
for (int i=1; i<10; i++)//Перебор всех спинБоксов
{
    QString SB_Name = "В"+QString::number(i); //формируем имя текущего спинБокса
    QSpinBox* CurrentB = QWidget::findChild <QSpinBox *> (SB_Name); // поиск нужного спинБокса по имени

    if (CurrentB->value()>0)//если в текущем спинБоксе >0
    {
        t++; //1,2
        if (t<=2) //если к этому моменту найдено не больше 2-ух ненулевых спинБоксов
        {
            switch (i) //установка минимального балла для текущего спинБокса
            {
                case 1: minP=40; //ИКТ
                    break;
                case 5: minP=37; //География
                    break;
                case 6: minP=22; //Ин.яз
                    break;
                case 7: minP=32; //История
                    break;
                case 8: minP=42; //Общага
                    break;
                case 9: minP=32; //Лит-ра
                    break;
                default:minP=36; // Все остальное:
                    break;    // Физика, Химия, Биология
            }
        }
    }
}
```

Если число в текущем поле больше, чем минимальный балл, найдем соответствующий ему лейбл. Текст, содержащийся в лейбле – название экзамена. Необходимо запомнить и название, и результат экзамена. Для этого используется объект `Inp` класса `Input` (информацию о классе см. в п. 3.3.2.).

Если в течение перебора полей ввода нашлись экзамены, результат которых был неудовлетворителен, или пользователь заполнил больше полей, чем допустимо, выводится соответствующее сообщение.

```
if (CurrentB->value()>=minP) // если введен балл >= минимуму то запомним его
{
    good++;
    Inp.extraP[good-1]= CurrentB->value();
    QString L_Name = "L"+QString::number(i);
    QLabel* CurrentL= QWidget::findChild <QLabel *> (L_Name);
    Inp.setName(CurrentL->text(),good-1);
}

} else
{
    ui->Message->setText("You chose more than 2 additional exams. Let only 2 of them and try again.");
    return; //принудительно выходим из обработчика
}

}
} //конец перебора

if ((t>0)&&(good==0)) //
{
    ui->Message->setText("Unfortunately, you failed additional exams. It is impossible to find an educational program for you.");
    return;
}
if (t==0)
{
    ui->Message->setText("You chose no additional exams. 1-2 exams are needed for search.");
    return;
}
```

Вывод таблицы:

После перебора полей ввода и заполнения полей объекта `Inp` посчитаем итоговую сумму баллов и создадим объект `File` для работы с `ege.txt`, и объект `U` класса `Uni` (информацию о классе см. в п. 3.3.3.).

Перед тем как приступить к обработке, нужно убрать ненужные элементы интерфейса и добавить те, с которыми предстоит работать.

```
Inp.setSum();
Uni U;
ui->tableWidget->show(); // делаем таблицу видимой
ui->AddButton->show(); //появляется кнопка для добавления записей

ui->Sum->show(); //появляется лейбл с суммой баллов
ui->Sum->setText("Total points: "+ QString::number(Inp.sum));

QFile File ("C:/Users/mi air/Desktop/kek/ege.txt");
File.open(QIODevice::ReadOnly);
```


В цикле (условие выхода: пока не достигнут конец файла) будем построчно читать ege.txt.

Метод `QFile::readLine` считывает из файла строку и возвращает ее в виде `QByteArray` (массив символов). Так как Qt не предусмотрено функций, переводящих `QByteArray` в `QString` (класс для работы со строками), используется следующий алгоритм:

- 1) `QByteArray` конвертируем в `std::string`;
- 2) `std::string` конвертируем в `QString`.

Далее, разделим считанную строку на секции. Каждой секции соответствует ячейка определенного столбца таблицы.

```
QFile File ("C:/Users/mi air/Desktop/kek/ege.txt");
File.open(QIODevice::ReadOnly);

while (!File.atEnd())
{
    std::string Line = File.readLine().toStdString(); //взяли строку и сконвертировали ее в std::string
    QString QL = QString::fromStdString(Line); //а теперь из std::string в QString

    U.name = QL.section(";", 0,0); //заполняем поля объекта для сравнения
    U.faculty = QL.section(";", 1,1);
    U.specialty = QL.section(";", 2,2);
    U.ExtExams[0] = QL.section(";", 3,3);
    U.ExtExams[1] = QL.section(";", 4,4);
    U.score = QL.section(";", 5,5);
    U.degree = QL.section(";", 6,6);
    U.comment = QL.section(";", 7,7);
}
```

Затем, чтобы определить, какие строки попадут в таблицу, необходимо сравнить содержимое объектов U и Inp. В U хранится информация из файла, а в Inp – информация, внесенная пользователем.

```
//сравним Input и Uni
if ((Inp.getName(0).toStdString()==U.ExtExams[0].toStdString())&&
    (Inp.getName(1).toStdString()==U.ExtExams[1].toStdString()))
{
    if (((U.comment == "---")&&(U.score.toInt()<=Inp.sum)) ||
        ((U.comment!="---")&&(U.score.toInt()-Inp.sum<=100)))
    {

```

Если совпали названия экзаменов, а сумма баллов равна или превышает проходной балл:

- 1) Добавим новую строку в таблицу;
- 2) В цикле переберем все столбцы. По номеру столбца определяется какая информация должна быть в текущей ячейке таблицы;
- 3) Используя класс QTableWidgetItem, создаем элемент таблицы и заполняем им соответствующее место в таблице.

Когда достигнут конец файла, и таблица заполнена, работу с ege.txt можно завершить. Закрываем файл, выполнение метода on_ButtonSearch_released завершено.

```
QString Info;
ui->tableWidget->setRowCount(ui->tableWidget->rowCount()+1); //добавляем строку
ui->tableWidget->setRowHeight(ui->tableWidget->rowCount()-1,200);
for (int col=0; col < ui->tableWidget->columnCount(); col++) //цикл по столбцам
{
    switch (col)
    {
        case 0: Info=U.name;
            break;
        case 1: Info=U.faculty;
            break;
        case 2: Info=U.specialty;
            break;
        case 3: Info="Russian Language\nMathematics\n"+U.ExtExams[0]+" \n"+U.ExtExams[1];
            break;
        case 4: Info=U.score;
            break;
        case 5: Info=U.degree;
            break;
        case 6: if (U.comment=="---") Info=U.comment;
            else Info=U.comment+"\nMinimum: "+QString::number(U.score.toInt()-Inp.sum);
            break;
    }
    QTableWidgetItem* itm = new QTableWidgetItem(Info);
    itm->setFlags(Qt::ItemIsEditable);
    itm->setTextAlignment(Qt::AlignCenter);
    ui->tableWidget->setItem(ui->tableWidget->rowCount()-1,col,itm);
}
}
}
File.close();
return;
}
```

3.3.1.4 Метод on_AddButton_released

Данный метод скрывает ненужные элементы интерфейса и показывает форму для добавления новой записи в ресурсный файл ege.txt, а также кнопку “Save”.

3.3.1.5 Метод on_SaveButton_released

Этот метод реализует занесение информации из формы (см. п. 3.3.1.4) в ресурсный файл.

Ниже, на рисунке 4, представлена блок-схема метода on_SaveButton_released.

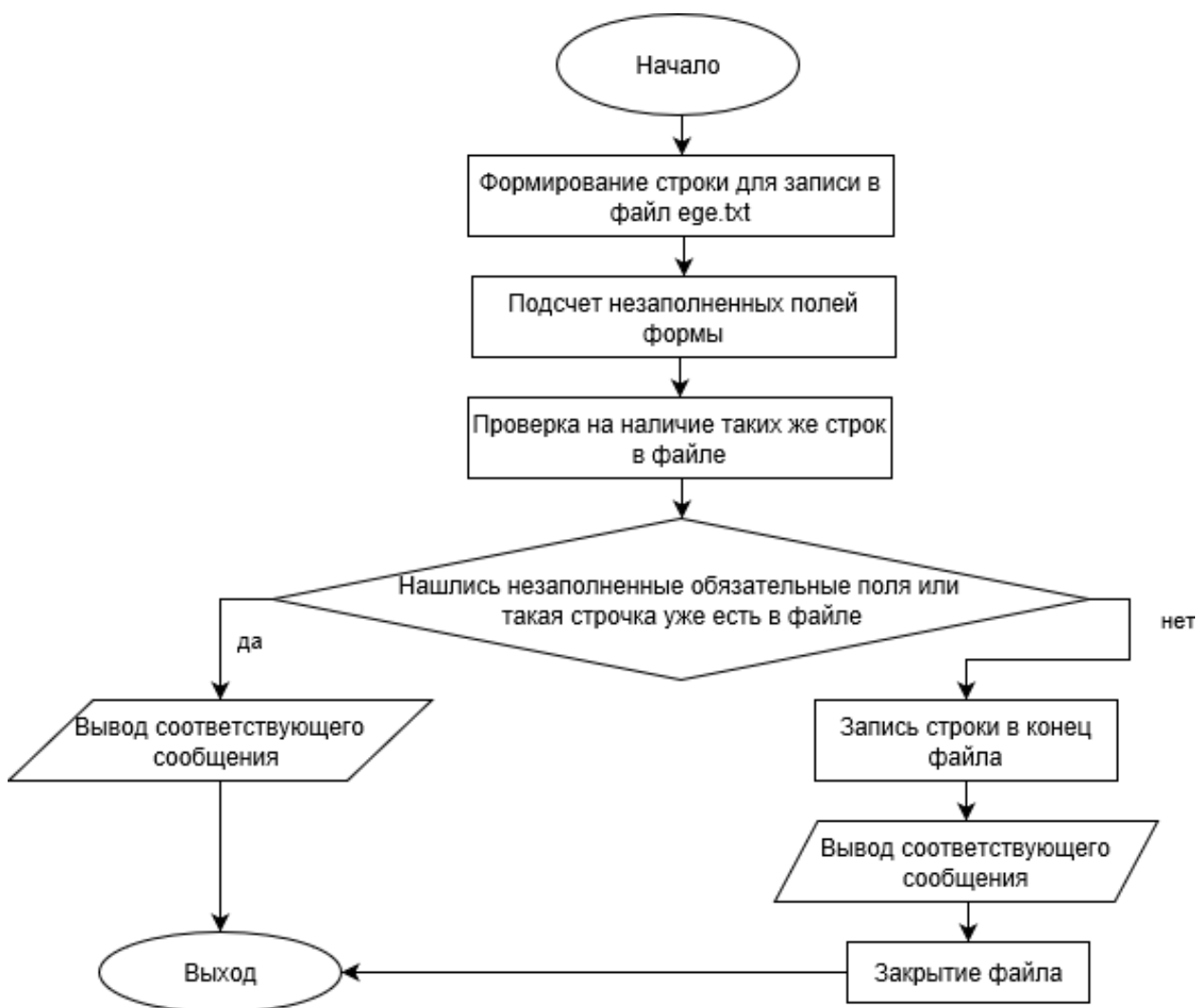


Рис. 4 Блок-схема метода on_SaveButton_released

3.3.2 Класс Input

Данный класс предназначен для сбора информации об экзаменах из формы и ее последующей обработки.

На рисунке 5 представлена UML-диаграмма класса Input.

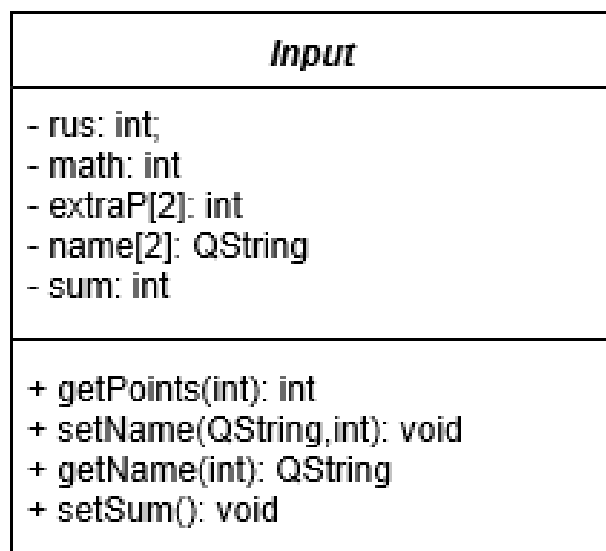


Рис. 5 UML-диаграмма класса Input

Поля `rus` и `math` содержат баллы за экзамены по Русскому языку и Математике, массив `extraP` – баллы, полученные за дополнительные экзамены. Массив `name` типа `QString` содержит названия дополнительных экзаменов. `Sum` – поле, предназначенное для хранения суммы баллов за 4 экзамена.

3.3.2.1 Метод `getPoints`

Метод возвращает значение поля с модификатором доступа `private`.

3.3.2.2 Метод `setName`

Метод принимает в качестве параметров строку `str` типа `QString` и число `index` типа `int` и устанавливает `str` как значение элемента массива `name [index]`.

3.3.2.3 Метод `getName`

Метод принимает в качестве параметра число `index` типа `int` и возвращает значение элемента массива `name[index]`.

3.3.2.4 Метод setSum

Метод вычисляет сумму баллов по 4 предметам: Русский, Математика и два дополнительных экзамена, и устанавливает ее как значение поля sum (рис. 6).

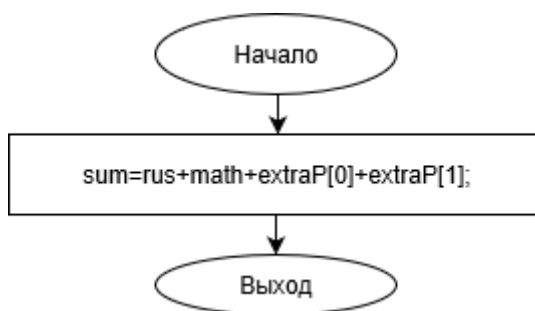


Рис.6 Блок-схема алгоритма вычисления суммы баллов sum

3.3.3 Класс Uni

Данный класс предназначен для работы с ресурсным файлом, в частности, для хранения разбитой на сегменты строки, прочитанной из ege.txt.

Все поля данного класса имеют модификатор доступа public. Класс не имеет ни одного метода.

Ниже, на рисунке 7, представлена UML-диаграмма класса Uni.

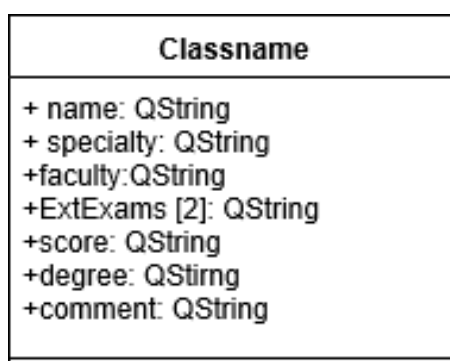


Рис.7 UML-диаграмма класса Uni

3.4 Описание графического интерфейса

Графический интерфейс приложения «USE Calculator» представлен на изображениях ниже.

При запуске приложения открывается стартовая страница (рис. 8), содержащая приветствие, краткое указание, форму для поиска и 2 кнопки: “Search”(Поиск) и “Add new table record” (Добавить новую запись).

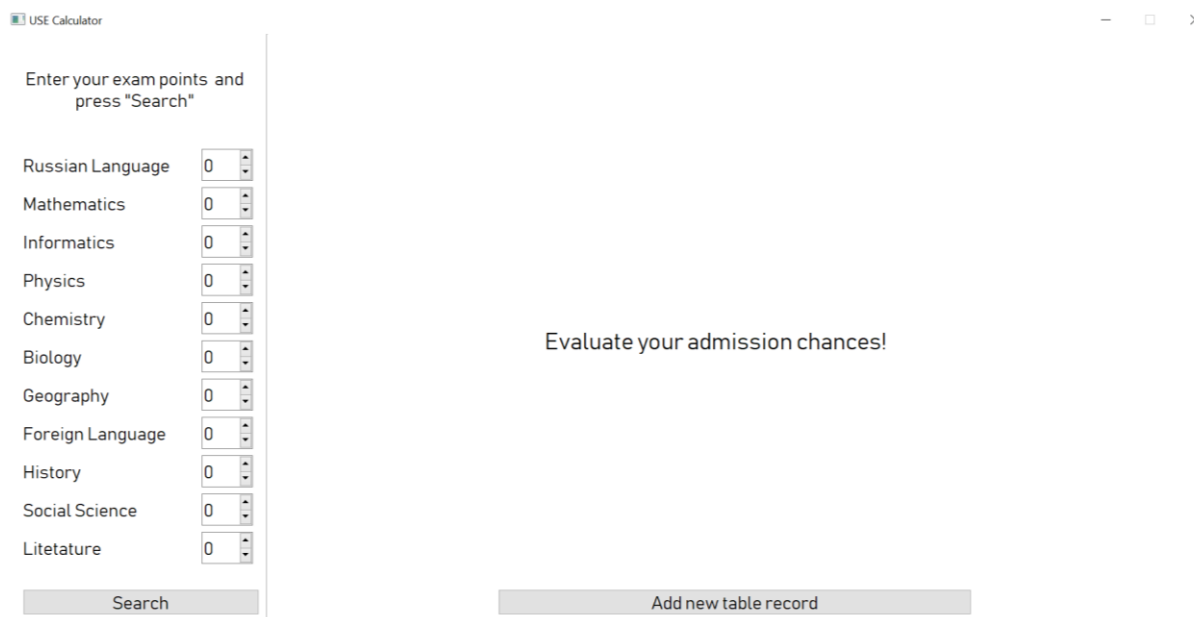


Рис. 8 Расположение элементов управления на форме

При нажатии пользователем кнопки поиска на форме появится таблица, содержащая информацию об образовательных программах университетов Москвы и вычисленная сумма баллов (рис.10), либо уведомление о проблеме (рис. 9).

USE Calculator

Enter your exam points and press "Search"

Russian Language 32

Mathematics 50

Informatics 40

Physics 0

Chemistry 0

Biology 0

Geography 0

Foreign Language 0

History 0

Social Science 0

Literature 0

Search

Unfortunately, you failed your exams. It is impossible to find an educational program for you.

Add new table record

Рис.9 Программа сообщает пользователю о невозможности подбора направления подготовки

USE Calculator

Enter your exam points and press "Search"

Russian Language 89

Mathematics 74

Informatics 94

Physics 0

Chemistry 0

Biology 0

Geography 0

Foreign Language 0

History 0

Social Science 0

Literature 0

Search

| University | Faculty | Specialty | USE | Points | Degree | Additional exams |
|-------------|-----------------------------------|--------------------------------------|--|--------|----------|------------------|
| Bauman MSTU | Informatics and Control Systems | Information Systems and Technologies | Russian Language Mathematics Informatics | 255 | Bachelor | --- |
| Bauman MSTU | Social and The Humanities | Applied Informatics | Russian Language Mathematics Informatics | 243 | Bachelor | --- |
| RTU MIREA | Information Technology Department | Software Engineering | Russian Language Mathematics Informatics | 242 | Bachelor | --- |
| RTU MIREA | Information Technology Department | Applied Mathematics and Informatics | Russian Language Mathematics Informatics | 243 | Bachelor | --- |

Total points: 257

Add new table record

Рис. 10 Таблица с результатами поиска

Таблица имеет семь столбцов: университет, факультет, специальность, перечень ЕГЭ, проходной балл на данную специальность, и ДВИ.

Предусмотрена функция добавления новой информации в таблицу.

При выборе пользователем действия “Add new table record” появляется форма (рис. 11) для внесения информации о специальности, которая не учтена приложением.

The screenshot shows a web application titled "USE Calculator". On the left, there is a sidebar with the heading "Enter your exam points and press 'Search'". Below this heading is a list of subjects, each with a corresponding input field for exam points (all currently set to 0): Russian Language, Mathematics, Informatics, Physics, Chemistry, Biology, Geography, Foreign Language, History, Social Science, and Literature. At the bottom of the sidebar is a "Search" button. The main area of the application is titled "Fill the information about educational program". It contains several input fields: University, Faculty, Specialty, Extra exams, Entrance points (with a small dropdown arrow), Degree, and Additional exam. At the bottom of the main area is a "Save" button.

Рис. 11 Форма для добавления новой записи в таблицу

Если пользователь внесет недостаточное количество информации (рис. 12), или такая запись уже есть в таблице, программа сообщит пользователю о невозможности добавления такой записи (рис. 13).

USE Calculator

Enter your exam points and press "Search"

Russian Language 0

Mathematics 0

Informatics 0

Physics 0

Chemistry 0

Biology 0

Geography 0

Foreign Language 0

History 0

Social Science 0

Litetature 0

Search

Fill the information about educational program

University RTU MIREA

Faculty

Specialty Applied Inpormatics

Extra exams

Entrance points

Degree

Additional exam

Save

Рис. 12 Внесение неполной информации в форму

USE Calculator

Enter your exam points and press "Search"

Russian Language 0

Mathematics 0

Informatics 0

Physics 0

Chemistry 0

Biology 0

Geography 0

Foreign Language 0

History 0

Social Science 0

Litetature 0

Search

Unfortunately, it is impossible to save this record.

Add new table record

Рис. 13 Результат обработки недостаточного количества информации

В противном случае, если форма заполнена корректно (рис. 14), пользователь увидит сообщение о том, что его запись успешно сохранена (рис. 15).

USE Calculator

Enter your exam points and press "Search"

Russian Language 0

Mathematics 0

Informatics 0

Physics 0

Chemistry 0

Biology 0

Geography 0

Foreign Language 0

History 0

Social Science 0

Litetature 0

Search

Fill the information about educational program

University RTU MIREA

Faculty Institute of Physics and Technology

Specialty Optotechnics

Extra exams Physics

Entrance points 191

Degree Bachelor

Additional exam ---

Save

Рис. 14 Пример правильного заполнения формы

USE Calculator

Enter your exam points and press "Search"

Russian Language 0

Mathematics 0

Informatics 0

Physics 0

Chemistry 0

Biology 0

Geography 0

Foreign Language 0

History 0

Social Science 0

Litetature 0

Search

The record was successfully saved.

Add new table record

Рис.15 Результат обработки корректно заполненной формы

Заключение

По итогам разработки курсовой работы были выполнены следующие цели:

1. Для приложения был спроектирован и разработан интерфейс, который доступно демонстрирует возможности, предоставляемые пользователю.
2. Все функции приложения, указанные в техническом задании, реализованы в полной мере и работают без ошибок.

Исходя из представленных тезисов, поставленную в начале данной работы цель можно считать выполненной.

Список литературы

1. Официальная документация Qt [Электронный ресурс].
URL: <https://www.qt.io/>

Приложение 1. Техническое задание

1.1. Введение

Составленное техническое задание по дисциплине “Объектно-ориентированное программирование” является документом к курсовой работе, который отображает этапы разработки, в том числе стадии проектирования и документирования программы.

1.1.1. Наименование программы

Название приложения “USE Calculator”. Название отражает суть приложения.

1.1.2. Краткая характеристика области применения программы

Программа предназначена для быстрого поиска образовательных программ в вузах Москвы на основе результатов ЕГЭ.

1.2. Основание для разработки

Задание на курсовую работу.

1.3. Назначение разработки

Назначением программного приложения является поиск образовательных программ в университетах Москвы на основе результатов ЕГЭ и выдача полной информации о направлении подготовки пользователю, добавление новых данных к уже имеющимся.

1.4. Требования, предъявленные к программе

1.4.1. Взаимодействие продукта с другими продуктами и компонентами

Приложение взаимодействует с ресурсным файлом ege.txt.

1.4.2. Функциональные требования

Приложение должно реализовать следующие функции:

- 1) Поиск и выдача информации о специальностях
- 2) Добавление новой информации

1.4.3. Требования к внешним интерфейсам

Требования к внешним интерфейсам не предъявляются.

1.4.4. Требования к производительности

Приложение должно работать стабильно, таким образом, чтобы действия пользователя не вызвали аварийного завершения программы.

1.4.5. Нефункциональные требования

Требования к допустимости и безопасности приложения не предъявляются.

1.5. Требования к программной документации

Пояснительная записка в соответствии со стандартами РТУ МИРЭА; Отчетная документация, составленная в соответствии с ГОСТ.

В процессе создания приложения вся проделанная работа документируется, должны быть отражены детали разработки. Всё вышеперечисленное должно быть отражено в пояснительной записке, которая прилагается к работе.

1.6. Техничко-Экономические показатели

Техничко-Экономические требования к приложению не предъявляются.

1.7. Стадии и этапы разработки

Этапы разработки:

1. Исследование темы; (20.02.19-21.02.19)
2. Выработка требований и составление ТЗ; (22.02.19-23.02.19)
3. Проектирование; (24.02.19-28.02.19)
4. Реализация компонента: написание логики программы;
(15.03.19-30.04.19)
5. Интеграция компонента: написание графического интерфейса;
(30.04.19-15.04.19)
6. Отладка компонента; (15.04.19-01.05.19)
7. Оформление отчётной документации; (02.05.19-10.05.19)
8. Предъявление работы к защите. (07.06.19)

Дальнейшее сопровождение программы после сдачи и защиты курсового проекта (работы) не предусмотрено.

1.8. Порядок контроля и приёма работы

Приём работы осуществляется путём защиты курсовой работы у преподавателя, назначенного руководителем курсовой работы. Защита осуществляется в соответствии с установленным графиком защиты курсовых работ.