


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

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 ИУ6-21Б

|   |                                    |
|---|------------------------------------|
| <br>(Подпись, дата) | Т.Е. Старжевский<br>(И.О. Фамилия) |
|---|------------------------------------|

Руководитель практики

|   |                                    |
|---|------------------------------------|
| <br>(Подпись, дата) | О.А. Веселовская<br>(И.О. Фамилия) |
|---|------------------------------------|

Оценка отлично

2022 г.

## З А Д А Н И Е на учебную практику

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

Студент группы ИУ6-21Б

Старжевский Тимофей Евгеньевич  
(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.

База данных (файл) трамвайного депо содержит сведения о подвижном составе: инвентарный номер вагона, модель, год производства, срок службы (лет). Программа должна в интерактивном режиме формировать файл, добавлять и удалять данные, а также воспринимать каждый из перечисленных запросов и давать на него ответ.

1. Найти модели трамваев, которые произведены начиная с заданного года.
2. Показать номера и модели трамваев, срок службы которых истек.
3. Определить количество трамваев заданной модели в депо.
4. Построить график количества произведенных трамваев по годам.

### **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.

Написать программу, помогающую запоминать слова иностранного языка. Программа предлагает слова, выбранные из некоторого множества случайным образом, а обучающийся должен ввести его эквивалент. По желанию пользователя предусмотреть смену языка. Придумать методику оценивания ответов и формирования итогового балла.

### 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.

База данных трамвайного депо содержит сведения о подвижном составе: инвентарный номер вагона, модель, год производства, срок службы (лет). Программа должна в интерактивном режиме формировать файл, добавлять и удалять данные, а также воспринимать каждый из перечисленных запросов и давать на него ответ.


1. Найти модели трамваев, которые произведены начиная с заданного года.
2. Показать номера и модели трамваев, срок службы которых истек.
3. Определить количество трамваев заданной модели в депо.
4. Построить график количества произведенных трамваев по годам.

*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 2022 г.

Руководитель практики

 07.02.22  
(Подпись, дата)

О.А. Вессловская  
(И.О. Фамилия)

Студент

 07/02/22  
(Подпись, дата)

Т.Е. Старжевский  
(И.О. Фамилия)

## Оглавление

|  |    |
|--|----|
| Задание 1. Создание программной системы на Object Pascal. ....                                     | 5  |
| Объектная декомпозиция программы: .....  | 5  |
| Диаграмма пользовательского интерфейса: .....  | 6  |
| Диаграмма классов интерфейсной и предметной области программы: .....                               | 7  |
| Полученные формы интерфейса: .....   | 8  |
| Диаграмма последовательности действий процедуры <i>TMain.Button1Click</i> : .....                  | 9  |
| Задание 2. Создание программной системы с элементарным интерфейсом консольного режима на C++ ..... | 10 |
| Структурная декомпозиция: .....  | 10 |
| Структурная схема: .....   | 11 |
| Код программы: .....   | 14 |
| Задание 3. Создание программной системы с Qt интерфейсом на C++ .....                              | 17 |
| Объектная декомпозиция программы: .....  | 17 |
| Диаграмма пользовательского интерфейса: .....  | 18 |
| Диаграмма классов интерфейсной и предметной области программы: .....                               | 19 |
| Полученные формы интерфейса: .....   | 20 |
| Диаграмма последовательности действий процедуры <i>Admin::on_pushButton_BACK_clicked()</i> : ....  | 21 |
| Заключение: .....  | 22 |

## Вариант 16.

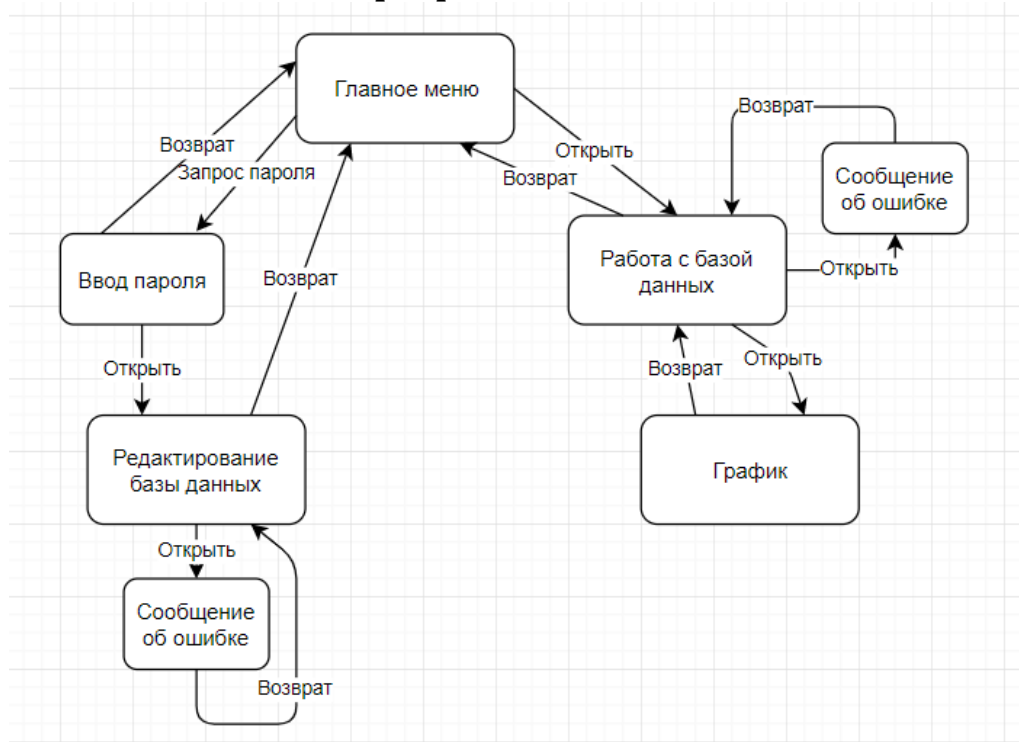
### Задание 1. Создание программной системы на Object Pascal.

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.

База данных (файл) трамвайного депо содержит сведения о подвижном составе: инвентарный номер вагона, модель, год производства, срок службы (лет). Программа должна в интерактивном режиме формировать файл, добавлять и удалять данные, а также воспринимать каждый из перечисленных запросов и давать на него ответ.

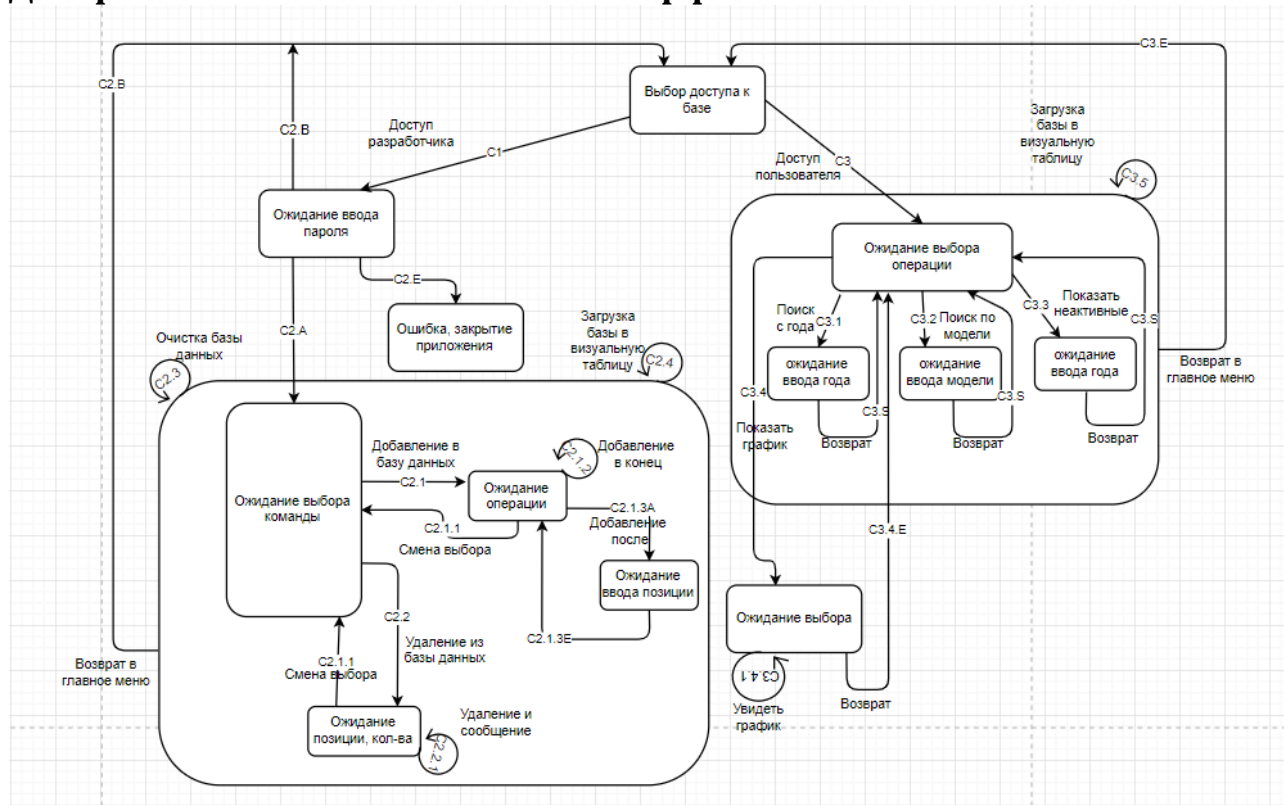
1. Найти модели трамваев, которые произведены начиная с заданного года.
2. Показать номера и модели трамваев, срок службы которых истек.
3. Определить количество трамваев заданной модели в депо.
4. Построить график количества произведенных трамваев по годам.

### Объектная декомпозиция программы:



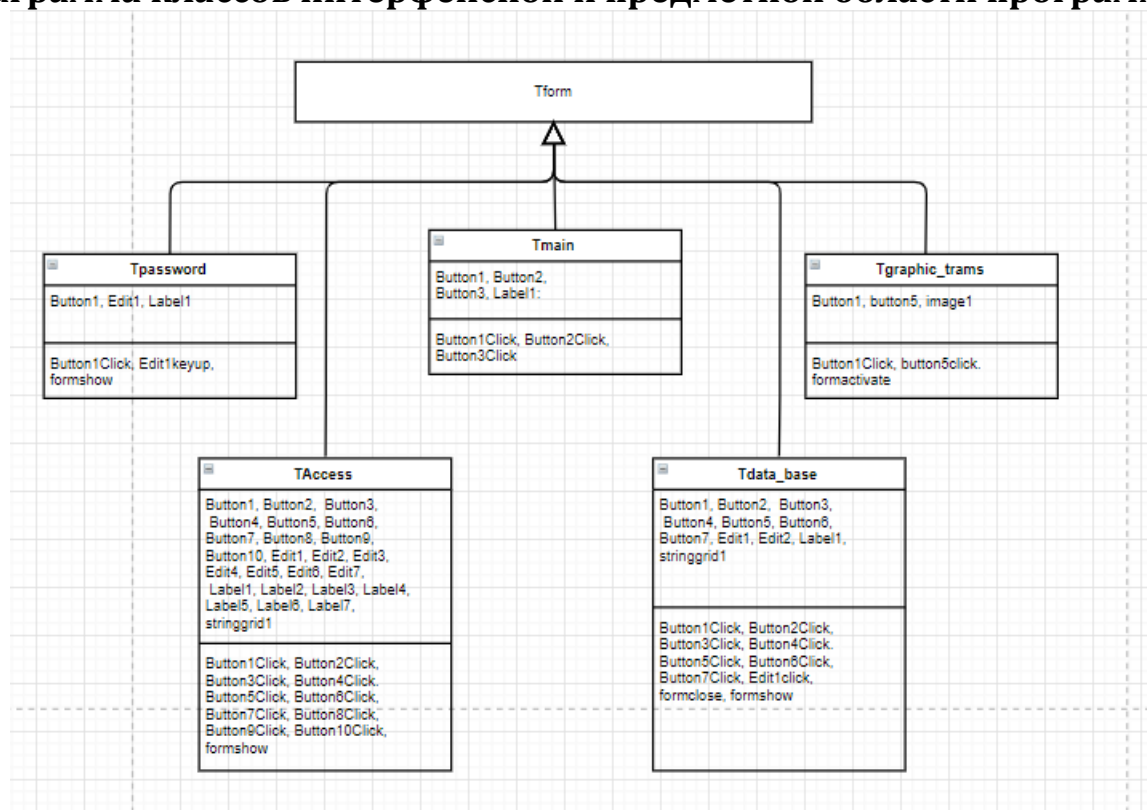


## Диаграмма пользовательского интерфейса:

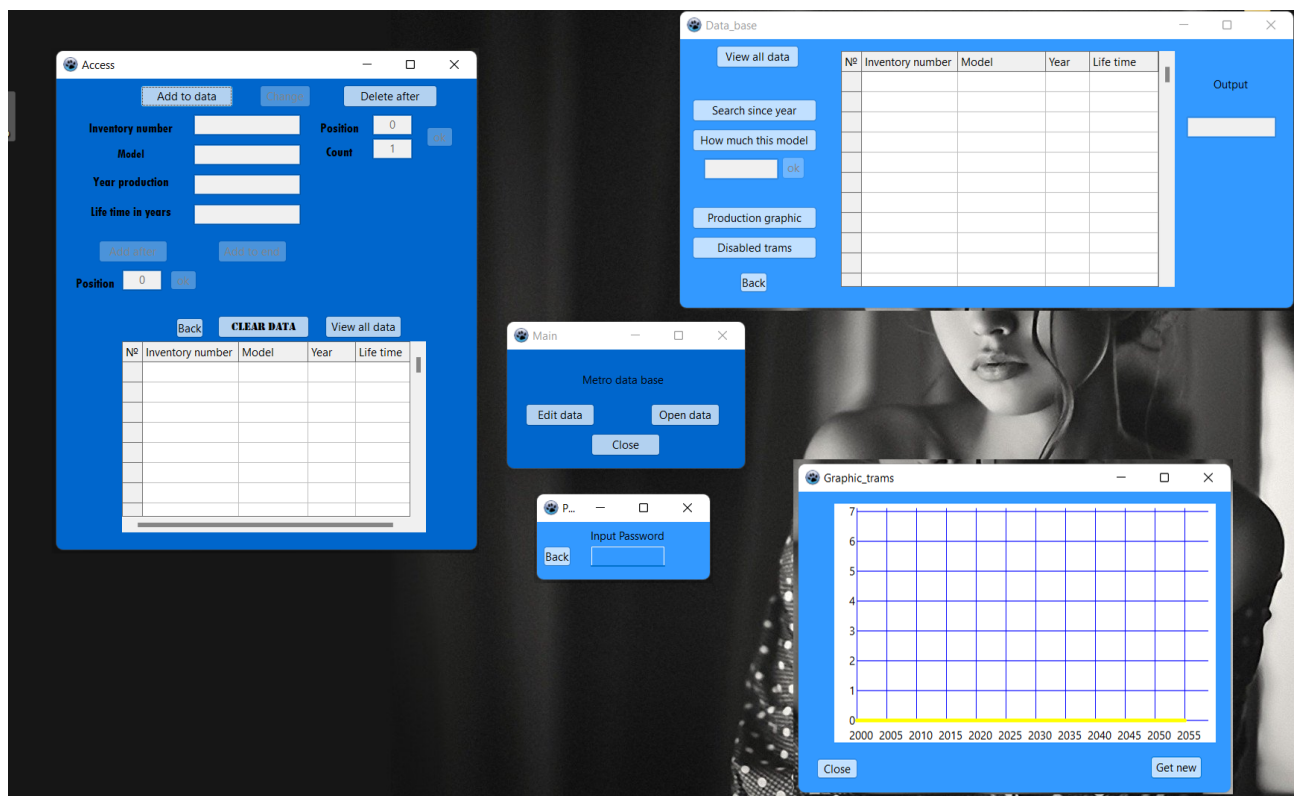
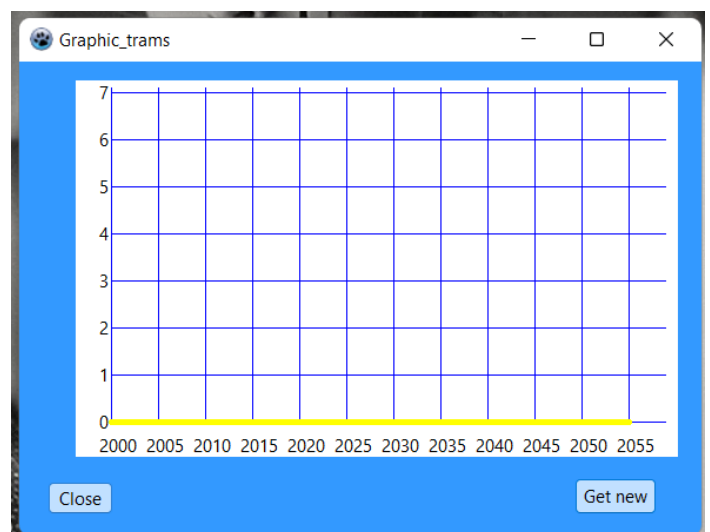
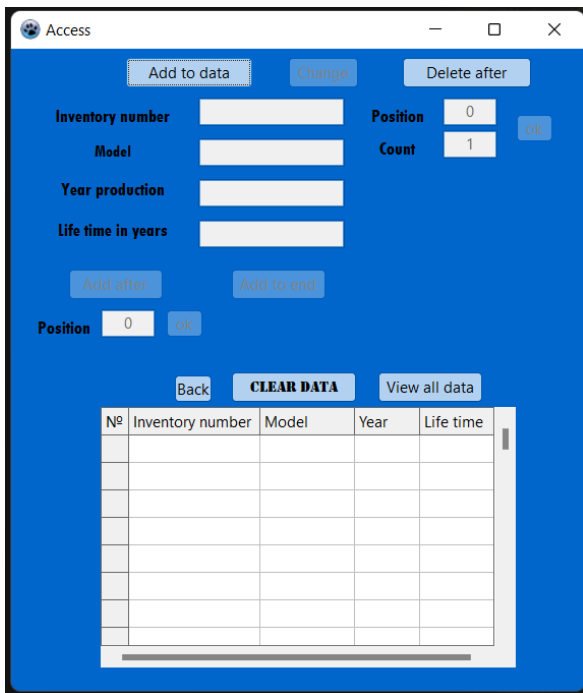
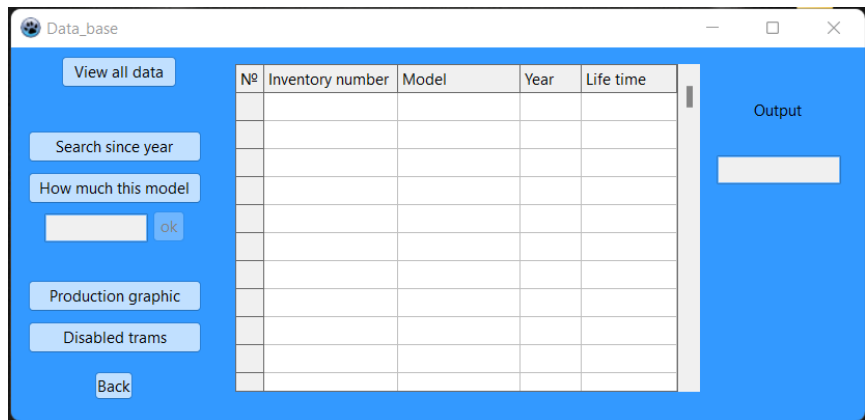
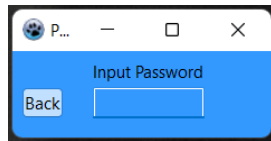
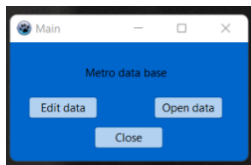


- C1 - нажатие кнопки "Edit Data"
- C2.A - ввод верного пароля в Edit1
- C2.E - ввод неверного пароля Edit1
- C2.B - Нажатие кнопки "Back"
- C2.E - Закрытие формы при превышении количества попыток
- C2.1 - Нажатие "Add to data"
- C2.1.1 - Нажатие "Change"
- C2.1.2 - Нажатие "Add to end"
- C2.1.3A - Нажатие "Add after"
- C2.1.3E - Нажатие "OK"
- C2.2 - Нажатие "Delete after"
- C2.2.1 - Нажатие "OK"
- C2.3 - Нажатие "Clear Data"
- C2.4 - Нажатие "View all data"
- C3 - Нажатие "Open data"
- C3.1 - Нажатие "Search since year"
- C3.2 - Нажатие "How much this model"
- C3.3 - Нажатие "Disabled trams"
- C3.4 - Нажатие "Production graphic"
- C3.S - Нажатие "Ok"
- C3.4E - Нажатие "Close"
- C3.4.1 - Нажатие "Get new"
- C3.5 - Нажатие "View all data"
- C3.E - Нажатие "Back"

## Диаграмма классов интерфейсной и предметной области программы:

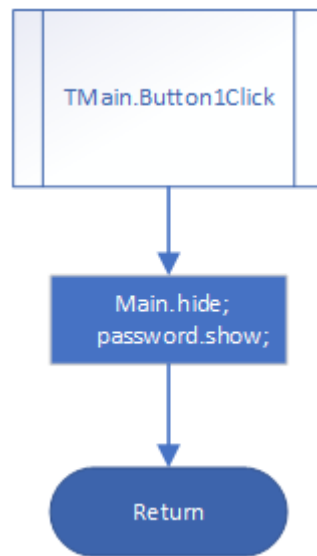


## Полученные формы интерфейса:





**Диаграмма последовательности действий процедуры**  
*TMain.Button1Click:*



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

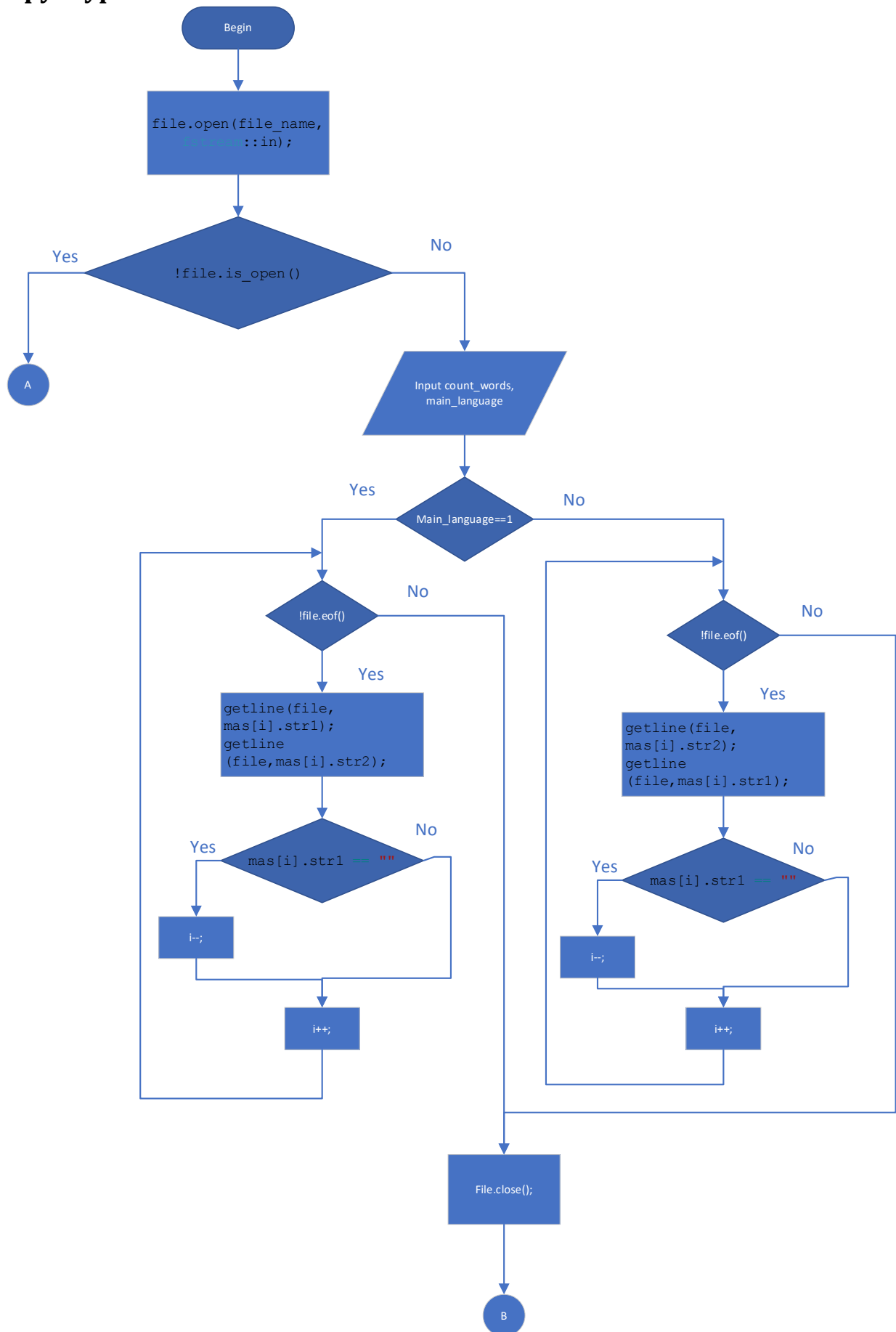
Выполнить структурную декомпозицию, разработать структурную схему, содержащую не менее 3 подпрограмм, и алгоритмы этих подпрограмм. Реализовать на C++ в консольном режиме. Предусмотреть примитивный интерфейс типа меню, позволяющий выбрать нужную подпрограмму.

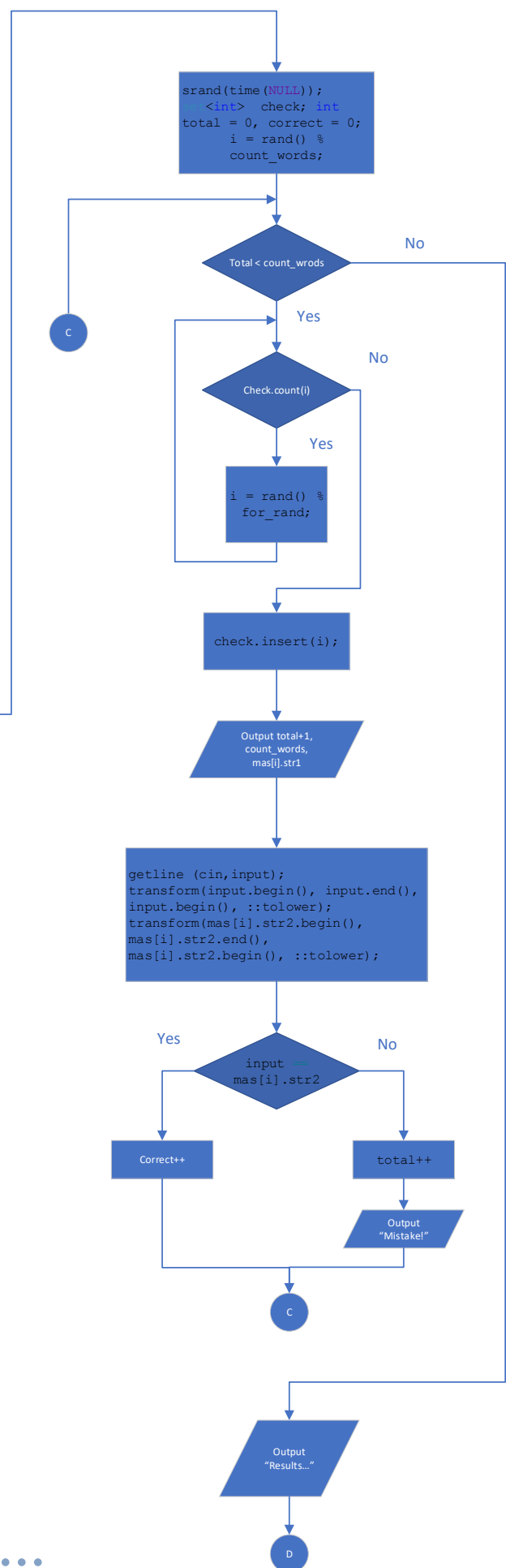
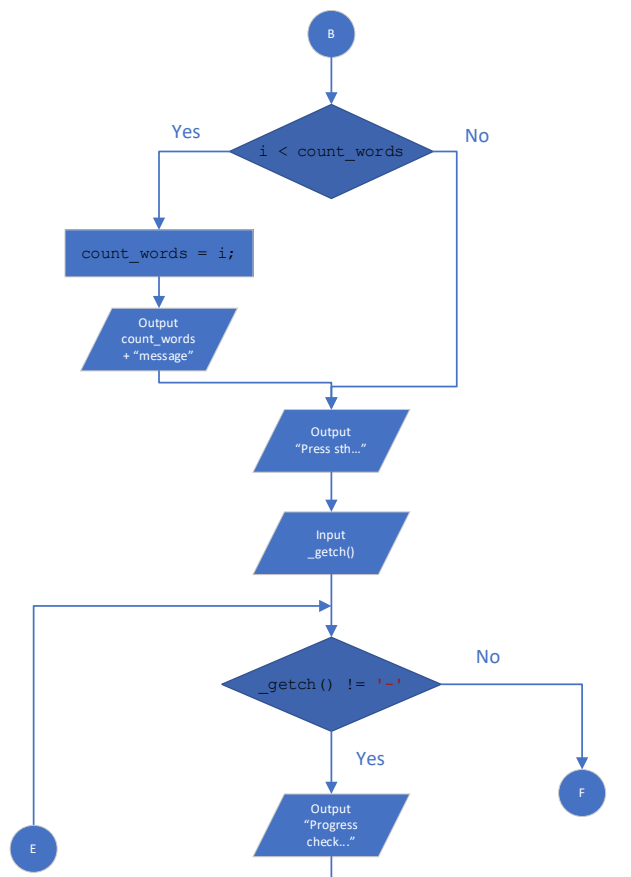
Написать программу, помогающую запоминать слова иностранного языка. Программа предлагает слова, выбранные из некоторого множества случайным образом, а обучающийся должен ввести его эквивалент. По желанию пользователя предусмотреть смену языка. Придумать методику оценивания ответов и формирования итогового балла.

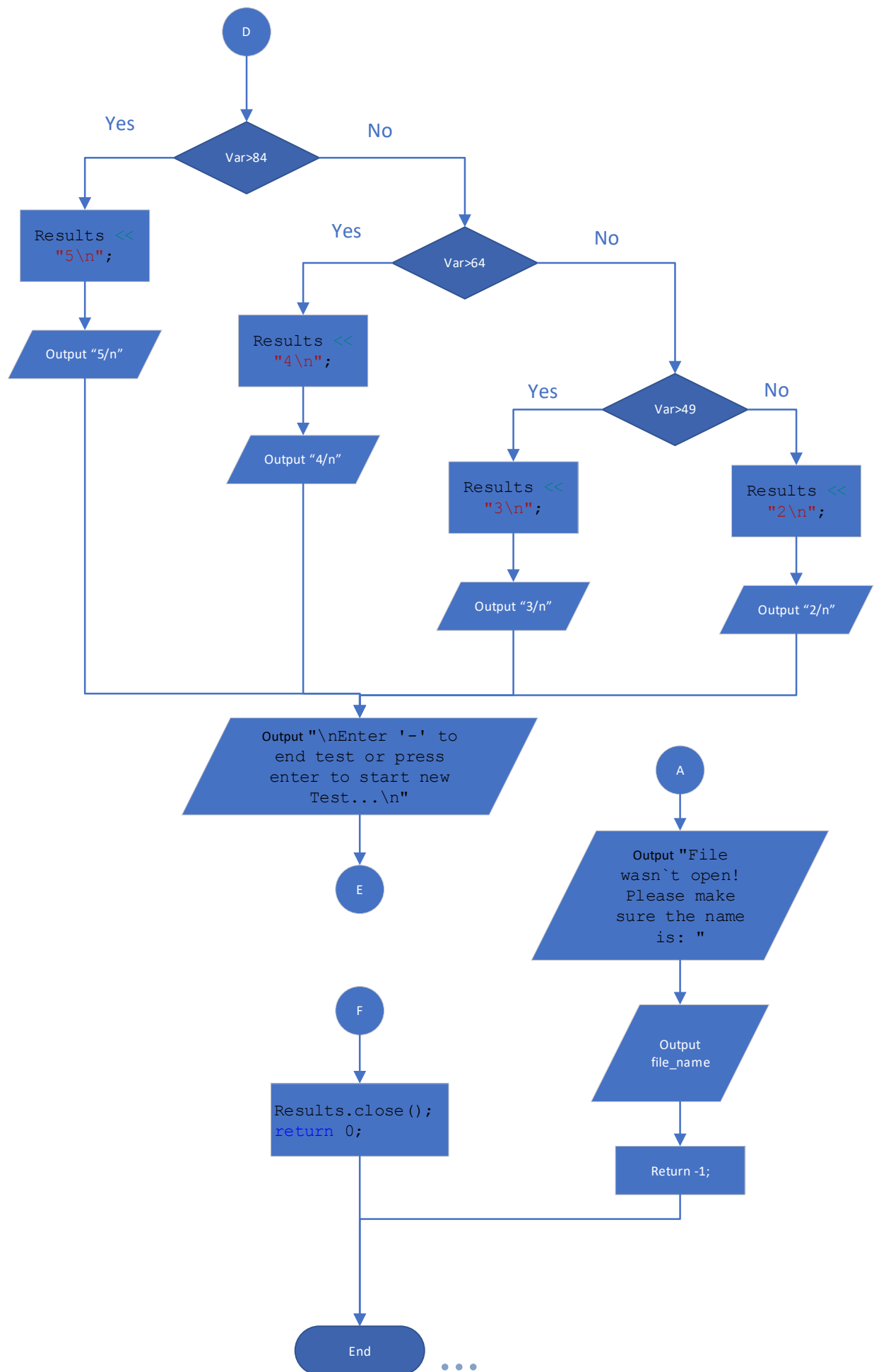
### Структурная декомпозиция:



## Структурная схема:







## Код программы:

```
#include <iostream>
#include <fstream>
#include <algorithm>
#include <time.h>
#include <set>
#include <conio.h>
#include <string>
// #define Teacher
// #define DEBUG
#define DEBUG_AFTER
#define DEBUG_CHEAT
// File format: "<English word>\n<Rus word>"
using namespace std;
struct slovo
{
    string str1, str2;
};
int main()
{
    string file_name = "Wordlist_5_1251.txt";
    system("chcp 1251 > nul");
    // system("chcp 1251");
#ifdef Teacher
    cout << "Input file name, standard it: " << file_name << "\nChange it?
[y/n]\t";
    char ch = _getch();
    if ( ch == 'y' || ch == 'Y' || ch == 'í')
    {
        cout << "Input new file name: "; cin >> file_name;
    }
    else cout << "\n Using standard file name (pls don't input 'Y' instead of 'y')
\n";
#endif
#ifdef DEBUG
    cout << "\nðàñðèððð ðóññèèè ýçûê\n";
#endif
    fstream file;
    file.open(file_name, fstream::in);
    if (!file.is_open())
    {
        cout << "File wasn't open! Please make sure the name is: " << file_name
<< endl;
        system("pause");
        return -1;
    }
    else
    {
        int count_words = 0, i = 0, main_language = 2;
        slovo mas[255]; string input;
        cout << "How much words u want to test (<256): ";
        cin >> count_words;
        cout << "Choose format:\n(1)Eng - rus || (2)Rus - eng \tMy choose: ";
        cin >> main_language;
        if (count_words > 255)
            count_words = 255;
        if (main_language == 1)
            while (!file.eof())
            {
                getline(file, mas[i].str1);
                getline (file, mas[i].str2); //str1 - eng str2 - rus
                if (mas[i].str1 == "") i--;
                i++;
            }
        else while (!file.eof())
        {
```



```

        getline(file, mas[i].str2);
        getline (file, mas[i].str1); //str1 - rus str2 - eng
        if (mas[i].str1 == "") i--;
        i++;
    }
    file.close();

    if (i < count_words)
    {
        count_words = i;
        cout << "File consist only " << count_words << " words, so let`s
start with it" << endl;
    }
    //Start counting
    cout << "Press something to start or '-' for end...\n";
    cin.ignore();
    int for_rand = i, tryes = 0;
    ofstream Results;
    Results.open("Results.txt");
    while (_getch() != '-')
    {
        cout << "===== Progress Check =====\n";
        srand(time(NULL)); set<int> check; int total = 0, correct = 0;
        i = rand() % count_words;
        while (total < count_words)
        {
            while (check.count(i))
                i = rand() % for_rand;
            check.insert(i);
            cout << "[" << total+1 << "/" << count_words << "]" " <<

mas[i].str1;
#ifdef DEBUG_CHEAT

            cout << "(" << mas[i].str2 << ")";

            cout << " - ";
            getline (cin, input);
            transform(input.begin(), input.end(), input.begin(),

::tolower);

            transform(mas[i].str2.begin(), mas[i].str2.end(),

mas[i].str2.begin(), ::tolower);
#ifdef DEBUG_AFTER

            cout << "Input: " << input << "\tCorrect: " << mas[i].str2

<< "\n";
#endif

            if (input == mas[i].str2)
                correct++;
            else
                cout << "Mistake! Try harder\n";
            total++;
        }
        int var = correct * 100 / total;
        cout << "==== Results =====\nTotal words: " << total <<
"\tCorrect translated: " << correct << "\tProcent: " << var << "\n--> Your mark: ";
        Results << ++tryes << ") Total words : " << total << "\tCorrect
translated : " << correct << "\tProcent : " << var << "\n--> Your mark : ";
        if (var > 84)
        {
            cout << "5\n";
            Results << "5\n";
        }
        else if (var > 64)
        {
            cout << "4\n";
            Results << "4\n";
        }
        else if (var > 49)

```

```

        {
            cout << "3\n";
            Results << "3\n";
        }
        else {
            cout << "2\n";
            Results << "2\n";
        }
        cout << "\nEnter '-' to end test or press enter to start new
Test...\n";

    }
    Results.close();
    return 0;
}

```

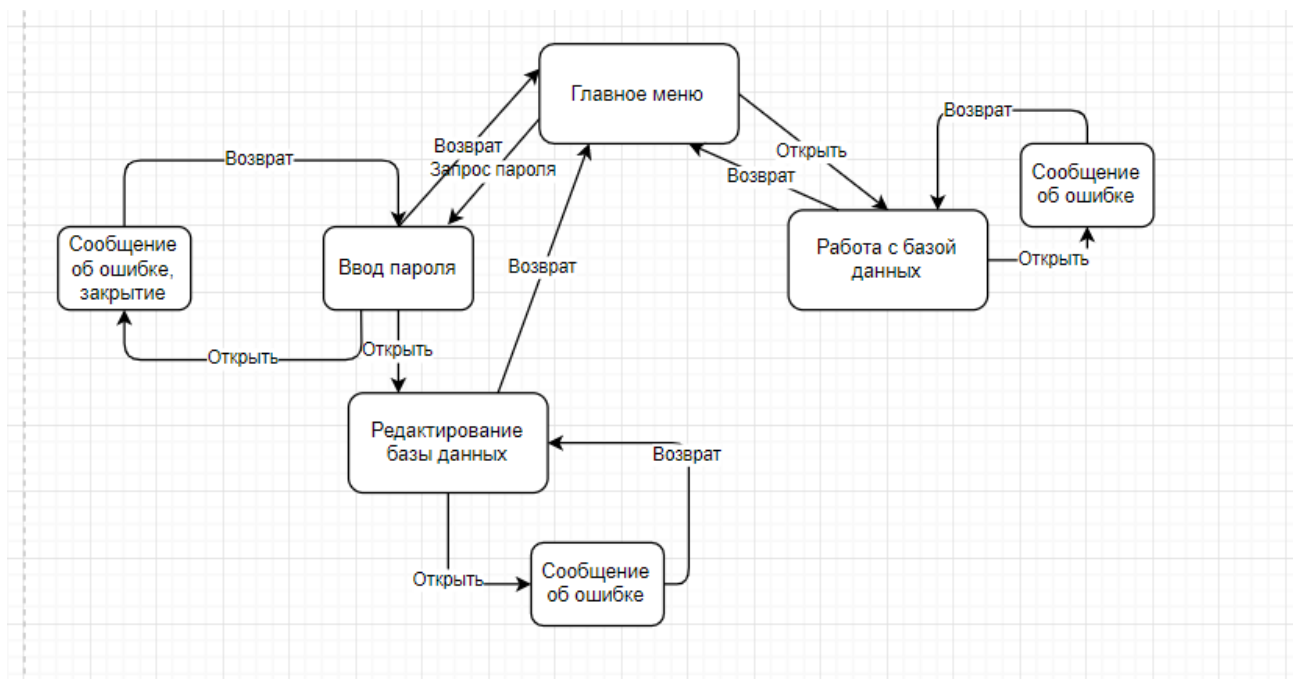
### Задание 3. Создание программной системы с Qt интерфейсом на C++

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.

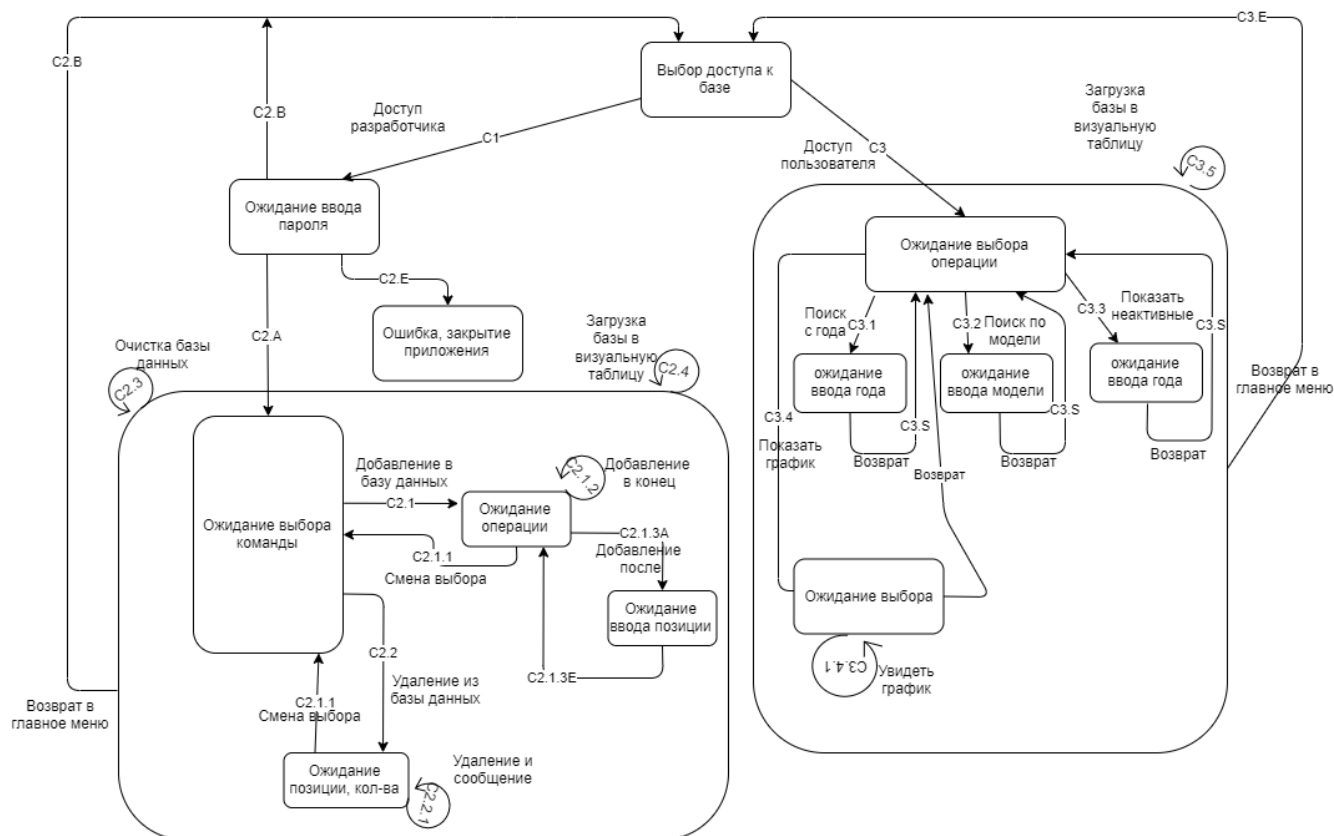
База данных трамвайного депо содержит сведения о подвижном составе: инвентарный номер вагона, модель, год производства, срок службы (лет). Программа должна в интерактивном режиме формировать файл, добавлять и удалять данные, а также воспринимать каждый из перечисленных запросов и давать на него ответ.

1. Найти модели трамваев, которые произведены начиная с заданного года.
2. Показать номера и модели трамваев, срок службы которых истек.
3. Определить количество трамваев заданной модели в депо.
4. Построить график количества произведенных трамваев по годам.

**Объектная декомпозиция программы:**

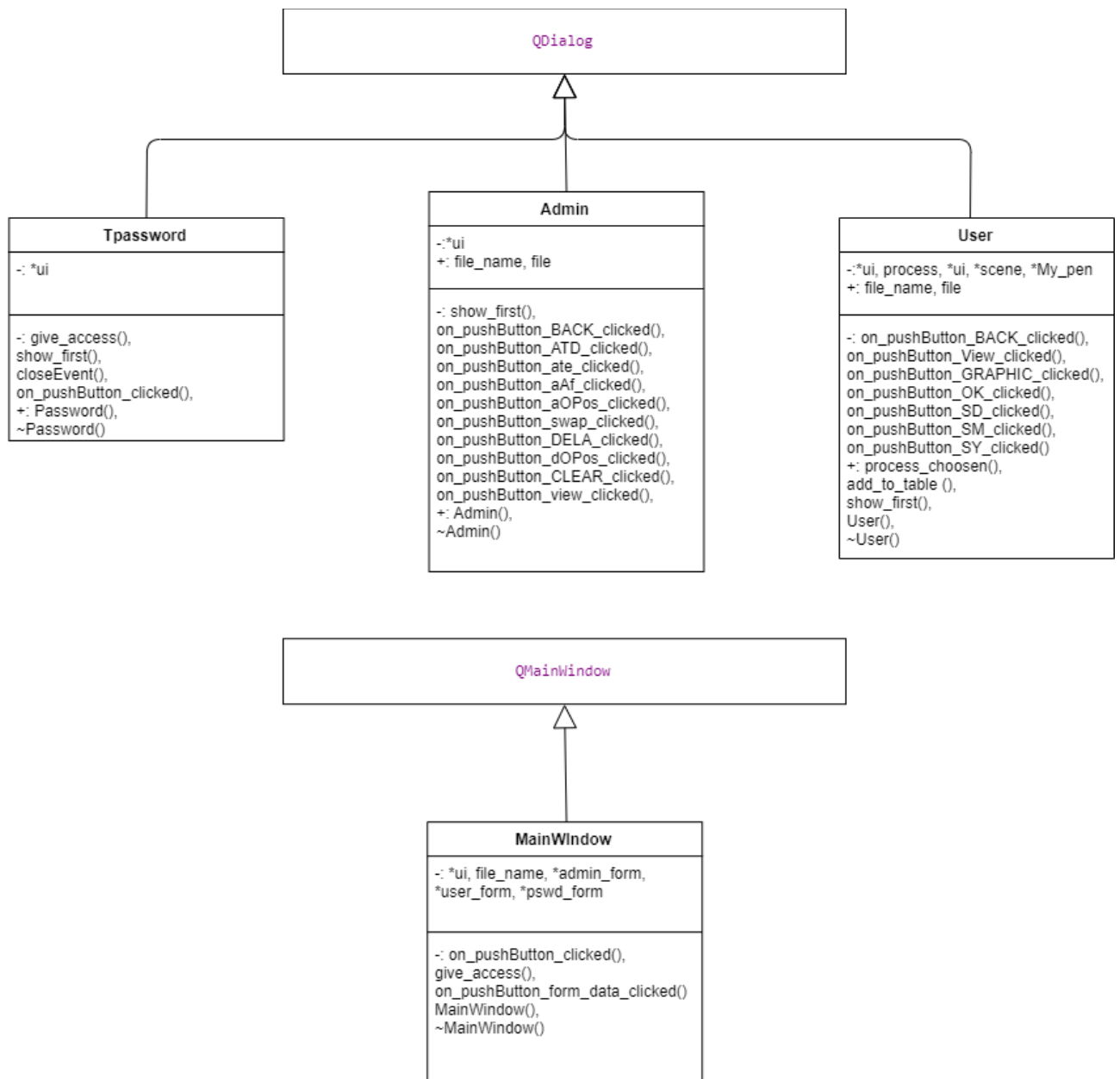


## Диаграмма пользовательского интерфейса:



- C1 - нажатие кнопки "Admin"
- C2.A - ввод верного пароля "Admin"
- C2.E - ввод неверного пароля в "Admin"
- C2.B - Нажатие кнопки "x"
- C2.E - Закрытие формы при превышении количества попыток
- C2.1 - Нажатие "Add to data"
- C2.1.1 - Нажатие "Change"
- C2.1.2 - Нажатие "Add to end"
- C2.1.3A - Нажатие "Add after"
- C2.1.3E - Нажатие "Ok"
- C2.2 - Нажатие "Delete after"
- C2.2.1 - Нажатие "Ok"
- C2.3 - Нажатие "Clear Data"
- C2.4 - Нажатие "View all data"
- C3 - Нажатие "User"
- C3.1 - Нажатие "Search since year"
- C3.2 - Нажатие "Models"
- C3.3 - Нажатие "Disabled"
- C3.4 - Нажатие "Production graphic"
- C3.S - Нажатие "Ok"
- C3.5 - Нажатие "View all data"
- C3.E - Нажатие "Back"

## Диаграмма классов интерфейсной и предметной области программы:



## Полученные формы интерфейса:

Dialog

Back to Main Change

Add to data Delete after

Inventory number: 001 Position: 0 ok

Model: BUGATTI Count: 0

Year production: 2022

Life time in years: 19

Add after Add to end

Position: 0 ok

|   | Model    | Inventory number | Year production | Life time |
|---|----------|------------------|-----------------|-----------|
| 1 | BENTLEY  | 001              | 2022            | 19        |
| 2 | MERCEDES | 001              | 2023            | 15        |
| 3 | TESLA    | 001              | 2020            | 10        |
| 4 | AUDI     | 001              | 2022            | 17        |
| 5 | TESLA    | 002              | 2020            | 10        |
| 6 | BENTLEY  | 0003             | 2022            | 8         |

View all data CLEAR DATA

MainWindow

Welcome to METRO database

Who are you:

Admin User

Dialog

Input password

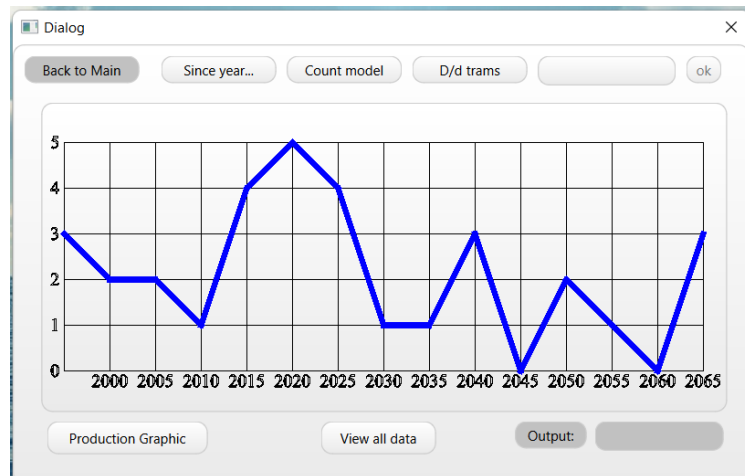
ok

Dialog

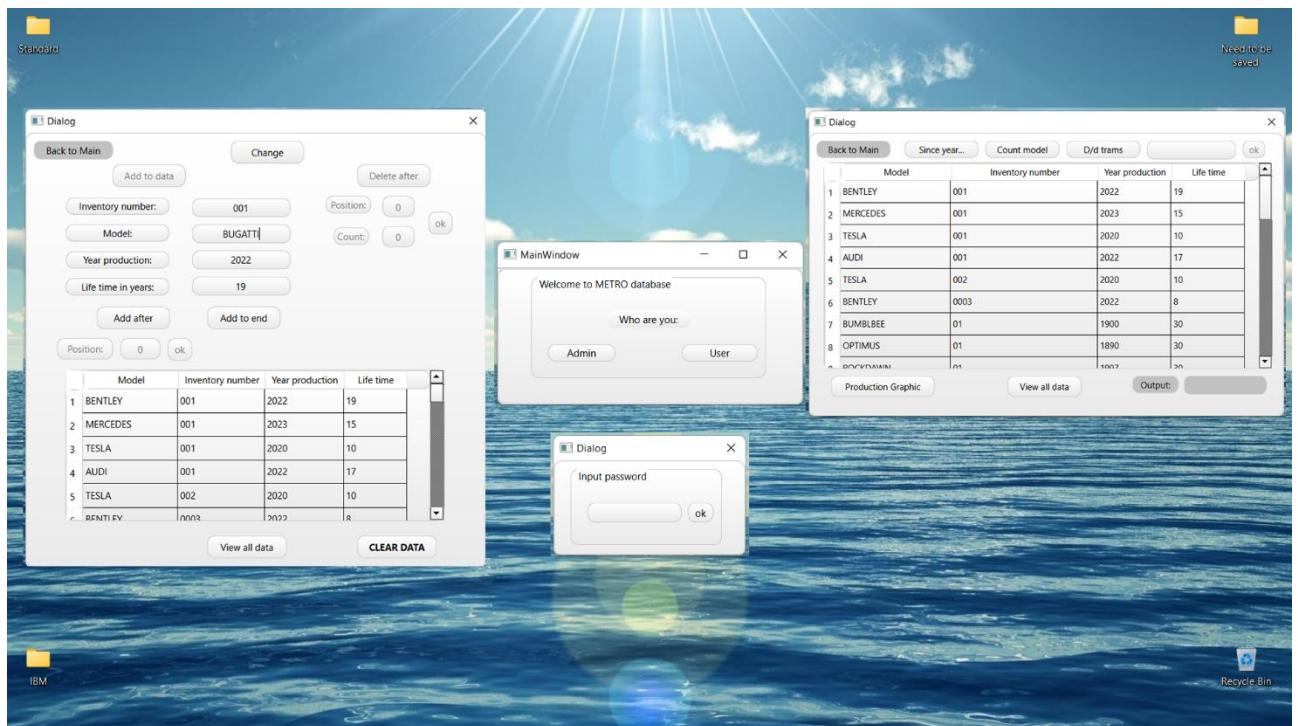
Back to Main Since year... Count model D/d trams ok

|   | Model    | Inventory number | Year production | Life time |
|---|----------|------------------|-----------------|-----------|
| 1 | BENTLEY  | 001              | 2022            | 19        |
| 2 | MERCEDES | 001              | 2023            | 15        |
| 3 | TESLA    | 001              | 2020            | 10        |
| 4 | AUDI     | 001              | 2022            | 17        |
| 5 | TESLA    | 002              | 2020            | 10        |
| 6 | BENTLEY  | 0003             | 2022            | 8         |
| 7 | BUMBLBEE | 01               | 1900            | 30        |
| 8 | OPTIMUS  | 01               | 1890            | 30        |
| 9 | ROCKDAWN | 01               | 1007            | 20        |

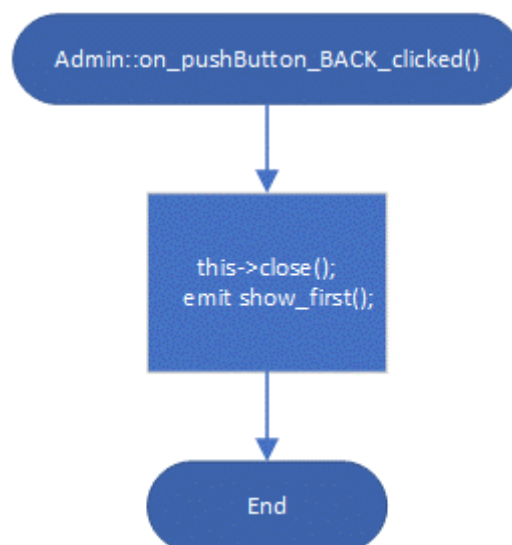
Production Graphic View all data Output:







**Диаграмма последовательности действий процедуры**  
*Admin::on\_pushButton\_BACK\_clicked():*



**Заключение:**

По заданным техническим заданиям было создано многооконное приложение для работы с базой данных в среде “Lazarus” и “QT” на разных языках программирования: Pascal и C++ соответственно. База данных хранится в файле, доступ к которому возможен с точки зрения администратора и пользователя. Работа с файлом протестирована и утечки данных внутри приложения не возникали. На языке C++ по второму техническому заданию была написана и протестирована программа для изучения русских/английских слов. При заданных условиях все работало корректно.