

Лабораторная работа по  
программированию  
"Электронный школьный дневник"

М.В.Булгакова

3 июня 2016 г.

# Глава 1

## Электронный школьный дневник

### 1.1 Введение

Школьный дневник, — основной документ школьника на время обучения. Дневник выполняет функции журнала регистрации оценок, полученных на уроках, замечаниях по поведению, средства общения учителей и родителей, а также показатель успеваемости ученика. В современном мире уже давно существует практика электронного школьного дневника. Данная лабораторная демонстрирует еще один вид электронного дневника школьника.

### 1.2 Задание

Создать проект "Электронный школьный дневник". Электронный школьный дневник - программа, позволяющая выставлять оценки за модуль, вычислять полугодовую и годовую оценки учащихся, просчитывать процент успеваемости класса, также реализовывающая поиск отличников, хорошистов и троечников.

### 1.3 Концепция

Программа должна предоставлять ученику/родителю возможность просмотра табеля данного ученика, процент его успеваемости. Учитель в отличие от ученика/родителя имеет права на изменение оценок, так же

может запросить список и количество отличников/хорошистов/троечников и процент успеваемости всего класса, так и отдельных учеников.

## 1.4 Минимально работоспособный продукт (MVP)

Программа, которая позволяет ученику/родителю возможность просмотра оценок данного ученика за год, и которая предоставляет учителю права на запрос список/количество отличников/хорошистов/троечников и процент успеваемости всего класса, так и отдельных учеников.

## 1.5 Диаграмма прецедентов использования

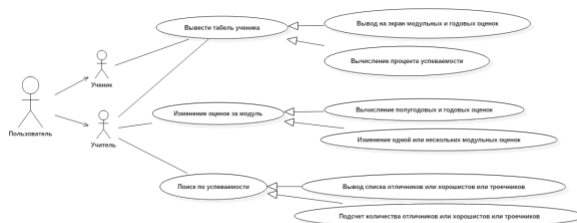


Рис. 1.1

На рис 1.1 изображена диаграмма прецедентов использования. Пользователь авторизуется вначале. И в зависимости от своего статуса ему предлагаются различные функции

## 1.6 Вывод

В данном разделе определены концепция готового приложения и MVP. Кроме того, в разделе представлена диаграмма прецедентов использования

## Глава 2

# Проектирование

### 2.1 Выделенные подпроекты

В процессе проектирования были выделены 4 подпроекта:

1. Core - Библиотека приложения
2. APP - Консольное приложение
3. Графическое приложение(нериализовано)
4. Тесты(нериализованы)

### 2.2 Описание элементов библиотеки

Интерфейс библиотеки содержит следующие методы:

```
void marks(student *);
```

- Метод, позволяющий найти среднюю оценку ученика, его успеваемость, качество выполнения работы

```
void find_excellent_pupil(student *);
```

- Метод, реализующий поиск отличников

```
void find_good_pupil(student *);
```

- Метод, реализующий поиск хорошистов

```
void find_lagging_pupil(student *);
```

- Метод, реализующий поиск отстающих учеников

```
void performance_calculation(student *);
```

- Метод, вычисляющий успеваемость всего класса, качество выполнения работы, среднюю оценку

## 2.3 Диаграмма компонентов

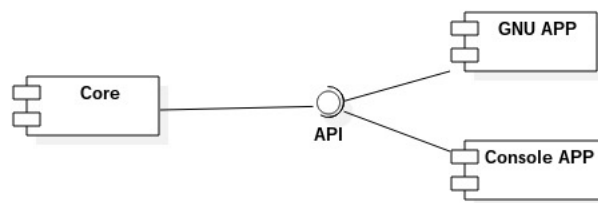


Рис. 2.1

На рис 2.1 представлена диаграмма компонентов, показывающая зависимости между основными компонентами приложения

## 2.4 Вывод

В данном разделе рассмотрен процесс проектирования приложения. Описаны выделенные подпроекты и методы интерфейса библиотеки, некоторые подпроекты не были реализованы из-за неправильного планирования разработки всего проекта.

# Глава 3

## Реализация приложения

### 3.1 Среда разработки

Операционная система: Debian(32-bit) Для создания проекта использовались Qt Creator 3.5.0 (opensource) и GCC.

### 3.2 Выделенные классы

В библиотеки выделено 3 класса:

1. Pupil - содержит основную информацию об ученике. Позволяет найти качество, успеваемость и среднюю оценку ученика.
2. Teacher - содержит основную информацию об учителе. Позволяет найти количество отличников, хорошистов, отстающих. Вычисляет качество, успеваемость, среднюю оценку всего класса.
3. CoreAPI - интерфейс ядра.

В подпроекте APP выделено 3 класса, обеспечивающих взаимодействия пользователя с ядром через консоль:

1. APupil
2. ATeacher
3. menu

### 3.3 Примеры работы консольного приложения

Для демонстрации работы консольного приложения ниже приведены снимки экрана работающего приложения.

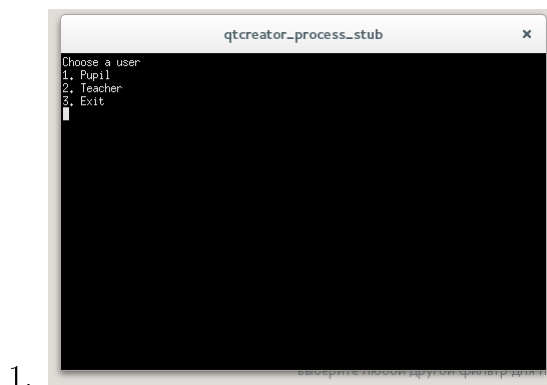


Рис. 3.1: рис1

На рис.1 представлено главное меню приложения

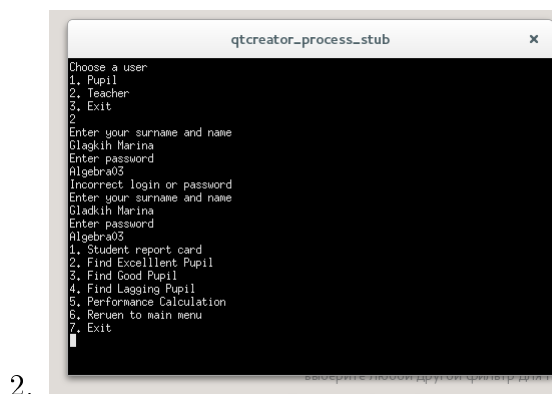


Рис. 3.2: рис2

На рис.2 показан пример аутентификации учителя и его меню

3.

```
qtcreator_process_stub
Incorrect login or password
Enter your surname and name
Gladkih Marina
Enter password
Algebra05
1. Student report card
2. Find Excellent Pupil
3. Find Good Pupil
4. Find Lagging Pupil
5. Performance Calculation
6. Return to main menu
7. Exit
2
Enter number of half-year
2
8. Ivanov Danila
9. Kistenev Anton
12. Masyuk Stanislav
14. Tkacheva Anastasia
Number of excellent pupil = 4
1. Return to Teacher's menu
2. Return to main menu
3. Exit
2
```

Рис. 3.3: рис3

На рис.3 представлен пример поиска отличников во втором полу-  
годии

4.

```
qtcreator_process_stub
Choose a user
1. Pupil
2. Teacher
3. Exit
1
Enter number of half-year
2
Enter your(student) surname and name
Bayborodin Nikita
5 Russian
5 Literature
4 Algebra
4 Geometry
5 History
5 Social Studies
4 Life Safety Fundamentals
5 PE
4 Biology
4 Geography
Progress 100 Quality 100 GPA 4
1. Return to Pupil's choice
2. Return to main menu
3. Exit
2
```

Рис. 3.4: рис4

На рис.4 представлен вход от имени ученика

## 3.4 Выводы

В данном разделе были описаны классы, выделенные при разработки приложения. Были сделаны снимки экрана, демонстрирующие работу консольного приложения



## Глава 4

# Обеспечение качества

### 4.1 Просмотр кода и демонстрации

В ходе разработки приложения были осуществлены 3 демонстрации, в ходе которых были выявлены ошибки работы программы и мелкие недочеты, были высказаны пожелания по функциональности. Все замечания были исправлены, а пожелания реализованы. Примером пожелания была кнопка выхода и возвращения в меню.

### 4.2 Тестирование

В ходе работы производилось ручное тестирование программы. Функциональные тесты, покрывающие часть ядра, были удалены, так как после изменение не компилировались, а на поиск ошибок времени не осталось.

### 4.3 Вывод

В данном разделе описаны демонстрации. Просмотр кода не осуществлялся, но автор понял, что пренебрегать им нельзя, так как он помогает выявить ошибки, незамеченные самим автором.

## Глава 5

### Выводы

Во время разработки приложения автор узнал возможности с++ и лучше усвоил принципы объектно-ориентированного программирования. Данная программа еще не закончена. Разработка графического интерфейса и покрытие тестами будет осуществляться до сентября текущего года.

## Листинги

```
1 #ifndef APUPIL_H
2 #define APUPIL_H
3 #include <iostream>
4 #include <string.h>
5 using namespace std;
6 #include "pupil.h"
7
8
9 class APupil
10 {
11 public:
12     APupil();
13     void pupil_mark();
14     void read(student *);
15     void aut(student *);
16     void print(student *);
17     void Dmenu();
18     void year_mark();
19     void ready(student *);
20
21 };
22
23 #endif // APUPIL_H
```

```
1 #include "apupil.h"
2 #include "pupil.h"
3 #include "menu.h"
4 #include <iostream>
5 #include <fstream>
6 #include <string>
7 #include <cstring>
8
9 using namespace std;
10 APupil::APupil()
11 {
12
13 }
```

```

14
15 void APupil::Dmenu()
16 {
17     puts("1. Return to Pupil's choice");
18     puts("2. Return to main menu");
19     puts("3. Exit");
20
21
22     int choice;
23     cin>>choice;
24     switch (choice) {
25     case 1:{
26         pupil_mark();
27         break;}
28     case 2:{
29         menu mmenu;
30         mmenu.main_menu();
31         break;}
32     case 3:{
33         cout<<"Good luck."<<endl;
34         break;}
35     }
36
37 }
38
39
40
41 void APupil::year_mark()
42 {
43     ifstream ProgressBook;
44     student st[30];
45     int i =st[1].num;
46     ProgressBook.open("../pupil1.txt");
47     for (i=1; i<16; i++){
48         ProgressBook>>st[i].surname>>st[i].name>>st[i].mark[
            russian]>>st[i].mark[literature]>>st[i].mark[algebra
            ]>>st[i].mark[geometry]>>st[i].mark[history]>>st[i].
            mark[SocialStudies]>>st[i].mark[LSF]>>st[i].mark[PE
            ]>>st[i].mark[biology]>>st[i].mark[geography];
49         ProgressBook.close();
50     }
51     ProgressBook.close();
52     ProgressBook.open("../pupil2.txt");
53
54     for (int i=15; i<31; i++){
55         ProgressBook>>st[i].surname>>st[i].name>>st[i].mark[
            russian]>>st[i].mark[literature]>>st[i].mark[
            algebra]>>st[i].mark[geometry]>>st[i].mark[history
            ]>>st[i].mark[SocialStudies]>>st[i].mark[LSF]>>st[

```

```

        i].mark[PE]>>st[i].mark[biology]>>st[i].mark[
        geography];
56     }
57     ProgressBook.close();
58     Pupil mPupil;
59     mPupil.year_marks(&st[30]);
60 }
61
62 void APupil::ready(student *st)
63 {
64
65     ifstream ProgressBook;
66     ProgressBook.open("../pupil3.txt");
67     int i=st[1].num;
68
69     for (i=1; i<16; i++){
70         ProgressBook>>st[i].surname>>st[i].name>>st[i].mark[
            russian]>>st[i].mark[literature]>>st[i].mark[
            algebra]>>st[i].mark[geometry]>>st[i].mark[history
            ]>>st[i].mark[SocialStudies]>>st[i].mark[LSF]>>st[
            i].mark[PE]>>st[i].mark[biology]>>st[i].mark[
            geography];
71     }
72     ProgressBook.close();
73
74
75 }
76
77 void APupil::pupil_mark()
78 {
79
80     student st[15];
81     read(&st[15]);
82     aut(&st[15]);
83     Dmenu();
84
85
86 }
87
88 void APupil::aut(student *st)
89 {
90     Pupil mPupil;
91     cout<<"Enter your(student) surname and name "<<endl;
92     cin>>mPupil.PupilSurname>>mPupil.PupilName;
93     int error;
94     for (int i=1; i<16; i++){
95         if ((strcmp(st[i].surname,mPupil.PupilSurname)==0)
            and(strcmp(st[i].name,mPupil.PupilName)==0)){
96             st[1].num=i;

```

```

97         mPupil.marks(&st[15]);
98         print(&st[15]);
99         i=45;
100        error=0;
101    }
102
103    else error=1;
104
105 }
106
107 if (error==1) cerr<<"Incorrect name or surname"<<endl;
108
109 }
110
111 void APupil::read(student *st)
112 {
113
114     ifstream ProgressBook;
115     cout<<"Enter number of half-year "<<endl;
116     int half_year;
117     cin>>half_year;
118     if (half_year==1) {ProgressBook.open("../pupil1.txt")
119         ;}
119     else
120         ProgressBook.open("../pupil2.txt");
121
122     for (int i=1; i<16; i++){
123         ProgressBook>>st[i].surname>>st[i].name>>st[i].mark[
            russian]>>st[i].mark[literature]>>st[i].mark[
            algebra]>>st[i].mark[geometry]>>st[i].mark[history
            ]>>st[i].mark[SocialStudies]>>st[i].mark[LSF]>>st[
            i].mark[PE]>>st[i].mark[biology]>>st[i].mark[
            geography];
124     }
125     ProgressBook.close();
126
127
128 }
129
130 void APupil::print(student *st)
131 {
132     int i=st[1].num;
133     cout<<st[i].mark[russian]<<" Russian"<<endl;
134     cout<<st[i].mark[literature]<<" Literature"<<endl;
135     cout<<st[i].mark[algebra]<<" Algebra"<<endl;
136     cout<<st[i].mark[geometry]<<" Geometry"<<endl;
137     cout<<st[i].mark[history]<<" History"<<endl;
138     cout<<st[i].mark[SocialStudies]<<" Social Studies"<<endl;
139     cout<<st[i].mark[LSF]<<" Life Safety Fundamentals"<<endl;

```

```

140     cout<<st[i].mark[PE]<<" PE"<<endl;
141     cout<<st[i].mark[biology]<<" Biology"<<endl;
142     cout<<st[i].mark[geography]<<" Geography"<<endl;
143     cout<<"Progress "<<st[i].Progress<<" Quality "<<st[i].
        Quality<<" GPA "<<st[i].GPA<<endl;
144 }

```

```

1  #ifndef ATEACHER_H
2  #define ATEACHER_H
3  #include <iostream>
4  #include<string.h>
5  using namespace std;
6  #include "pupil.h"
7
8  class ATeacher
9  {
10 public:
11     ATeacher();
12
13     void teacher_func();
14     void eu_main();
15     void d_main();
16     void pupil_marks();
17     void excelllent_pupil();
18
19     void good_pupil();
20     void lagging_pupil();
21     void performance();
22 };
23
24 #endif // ATEACHER_H

```

```

1  #include "ateacher.h"
2  #include "apupil.h"
3  #include "teacher.h"
4  #include "menu.h"
5  #include <iostream>
6  #include <fstream>
7  #include <string>
8  #include <cstring>
9  using namespace std;
10
11 ATeacher::ATeacher()
12 {
13
14 }
15
16 void ATeacher::teacher_func()
17 {

```

```

18     eu_main();
19 }
20
21 void ATeacher::d_main()
22 {
23     puts("1. Return to Teacher's menu");
24     puts("2. Return to main menu");
25     puts("3. Exit");
26
27
28     int choice;
29     cin>>choice;
30     switch (choice) {
31     case 1:{
32         eu_main();
33         break;}
34     case 2:{
35         menu mmenu;
36         mmenu.main_menu();
37         break;}
38     case 3:{
39         cout<<"Good luck."<<endl;
40         break;}
41     }
42 }
43
44 void ATeacher::eu_main()
45 {
46
47     char PassChek[20];
48
49     char TeacherSurname[20];
50     char TeacherName[20];
51     char password[20];
52
53
54     cout<<"Enter your surname and name "<<endl;
55     cin>>TeacherSurname>>TeacherName;
56
57     cout<<"Enter password"<<endl;
58     cin>>password;
59
60     ifstream TeacherBook;
61     TeacherBook.open("../teacher.txt");
62
63     int i,error=0;
64     char s1[20];
65     char s2[20];
66

```



```

67     for (i=1; i<10; i++){
68         TeacherBook>>s1>>s2>>PassChek;
69         if ((strcmp(s1,TeacherSurname)==0) and(strcmp(s2,
            TeacherName)==0) and (strcmp(password,PassChek)
            ==0)){
70             puts("1. Student report card");
71             puts("2. Find Excelllent Pupil");
72             puts("3. Find Good Pupil");
73             puts("4. Find Lagging Pupil");
74             puts("5. Performance Calculation");
75             puts("6. Reruen to main menu");
76             puts("7. Exit");
77
78
79             int choice;
80             cin>>choice;
81             switch (choice) {
82                 case 1:{
83                     pupil_marks();
84                     break;}
85                 case 2:{
86                     excelllent_pupil();
87                     break;}
88                 case 3:{
89                     good_pupil();
90                     break;}
91                 case 4:{
92                     lagging_pupil();
93                     break;}
94                 case 5:{
95                     performance();
96                     break;}
97
98                 case 6:{
99                     menu mmenu;
100                     mmenu.main_menu();
101
102                     break;}
103                 case 7:{
104                     cout<<"Good luck."<<endl;
105                     break;}
106             }
107             i=45;
108             error=0;
109         }
110         else error=1;
111     }
112     if (error==1){ cerr<<"Incorrect login or password"<<endl;
        eu_main();};

```

```

113
114
115     TeacherBook.close();
116 }
117
118 void ATeacher::pupil_marks()
119 {
120     APupil mPupil;
121
122     mPupil.pupil_mark();
123     d_main();
124
125 }
126
127 void ATeacher::excelllent_pupil()
128 {
129     Teacher mTeacher;
130     APupil mPupil;
131     student st[15];
132     mPupil.read(&st[15]);
133     mTeacher.find_excelllent_pupil(&st[15]);
134     d_main();
135
136 }
137
138 void ATeacher::good_pupil()
139 {
140     Teacher mTeacher;
141     student st[15];
142     mTeacher.find_good_pupil(&st[15]);
143     d_main();
144 }
145
146 void ATeacher::lagging_pupil()
147 {
148     Teacher mTeacher;
149     APupil mPupil;
150     student st[15];
151     mPupil.read(&st[15]);
152     mTeacher.find_lagging_pupil(&st[15]);
153     d_main();
154 }
155
156 void ATeacher::performance()
157 {
158     Teacher mTeacher;
159     APupil mPupil;
160     student st[15];
161     mPupil.read(&st[15]);

```

```

162     mTeacher.performance_calculation(&st[15]);
163     cout<<"Quality " <<st[1].Quality<<"% " <<" Progress " <<st
        [1].Progress<<"%"<<" GPA=" <<st[1].GPA<<endl;
164     d_main();
165 }

```

```

1  #include <iostream>
2  #include <fstream>
3  #include <string>
4  #include "menu.h"
5
6  using namespace std;
7
8  int main()
9  {
10     menu mMenu;
11     mMenu.main_menu();
12 }

```

```

1  #ifndef MENU_H
2  #define MENU_H
3
4
5  class menu
6  {
7  public:
8     menu();
9
10    void main_menu();
11 };
12
13 #endif // MENU_H

```

```

1  #include "menu.h"
2  #include "apupil.h"
3  #include "ateacher.h"
4  #include <iostream>
5  #include <fstream>
6  #include <string>
7
8
9  using namespace std;
10 menu::menu()
11 {
12
13 }
14
15 void menu::main_menu()

```

```

16 {
17     APupil mPupil;
18     ATeacher mTeacher;
19     puts("Choose a user");
20     puts("1. Pupil");
21     puts("2. Teacher");
22     puts("3. Exit");
23     int choice;
24     cin>>choice;
25     switch (choice) {
26     case 1:{
27         mPupil.pupil_mark();
28         break;}
29     case 2:{
30         mTeacher.teacher_func();
31         break;}
32     case 3:{
33         cout<<"Good luck!"<<endl;
34     }
35     }
36 }

```

```

1 #ifndef PUPIL_H
2 #define PUPIL_H
3 #include <iostream>
4 #include<string.h>
5 using namespace std;
6
7 struct student {
8     char name[20];
9     char surname[20];
10    int mark[10];
11    float oценка;
12    int Quality;
13    int Progress;
14    int num;
15    float GPA;
16 };
17
18 enum subject {russian,
19               literature,
20               algebra,
21               geometry,
22               history,
23               SocialStudies,
24               LSF,
25               PE,
26               biology,
27               geography};

```

```

28
29 class Pupil
30 {
31 public:
32     Pupil();
33     ~Pupil();
34     void marks(student *);
35     char PupilSurname[20];
36     char PupilName[20];
37     int half_year;
38     void year_marks(student *);
39 };
40
41 #endif // PUPIL_H

```

```

1 #include "pupil.h"
2 #include <iostream>
3 #include <fstream>
4 #include <string>
5 #include <cstring>
6 #include <cmath>
7
8 using namespace std;
9
10 Pupil::Pupil()
11 {
12
13 }
14
15 void Pupil::year_marks(student *st)
16 {
17     ofstream fout("../pupil3.txt");
18     for (int i=1; i<16; i++){
19         for (int j=0; j<11; j++){
20             st[i].mark[j]=floor((st[i].mark[j]+st[i+16].mark[j]
21                                     ))/2);
22             fout<<st[i].surname<<" "<<st[i].name<<" "<<st[i].mark[
23                 russian]<<" "<<st[i].mark[literature]<<" "<<st[i].
24                 mark[algebra]<<" "<<st[i].mark[geometry]<<" "<<st[i]
25                 .mark[history]<<" "<<st[i].mark[SocialStudies]<<"
26                 "<<st[i].mark[LSF]<<" "<<st[i].mark[PE]<<" "<<st[i]
27                 .mark[biology]<<" "<<st[i].mark[geography]<<endl;
28         }
29     }
30     fout.close();
31 }
32
33 void Pupil::marks(student *st)
34 {

```

```

29         int i=st[1].num;
30         int j;
31         int k2=0, k4=0, k5=0, summ=0;
32         for (j=0; j<10; j++){
33             summ=summ+st[i].mark[j];
34             if (st[i].mark[j]==2) k2++;
35             if (st[i].mark[j]==4) k4++;
36             if (st[i].mark[j]==5) k5++;
37         }
38
39         st[i].Quality=(k4+k5)*10;
40         st[i].Progress=(10-k2)*10;
41         st[i].GPA=summ/10;
42
43     }
44
45
46
47 Pupil::~~Pupil()
48 {
49
50 }

```

```

1  #ifndef TEACHER_H
2  #define TEACHER_H
3  #include "pupil.h"
4
5
6  class Teacher
7  {
8  public:
9      Teacher();
10     ~Teacher();
11     void find_excellent_pupil(student *);
12     void find_good_pupil(student *);
13     void find_lagging_pupil(student *);
14     void performance_calculation(student *);
15     void change_mark(student *);
16
17
18
19
20 };
21
22 #endif // TEACHER_H

```

```

1  #include "teacher.h"
2  #include "pupil.h"
3  #include <iostream>

```

```

4 #include <fstream>
5 #include <string>
6 #include <cstring>
7 using namespace std;
8
9 Teacher::Teacher()
10 {
11
12 }
13
14
15
16
17 void Teacher::find_excelllent_pupil(student *st)
18 {
19
20     int i;
21     int k=0;
22
23     for (i=1; i<16; i++){
24         int j;
25         int k5=0;
26         for (j=0; j<10; j++){
27             if (st[i].mark[j]==5) k5++;
28         }
29         if (k5==10) {
30             cout<<i+1<<". " <<st[i].surname<< " " <<st[i].name<<
31                 endl;
32             k=k+1;
33         }
34
35     }
36     cout<<"Number of excelllent pupil = " <<k<<endl;
37 }
38
39 void Teacher::find_good_pupil(student *st)
40 {
41
42     int i;
43     int k=0;
44
45     for (i=1; i<16; i++){
46         int j;
47         int k5=0, k3=0, k2=0;
48         for (j=0; j<10; j++){
49             if (st[i].mark[j]==5) k5++;
50             if (st[i].mark[j]==3) k3++;
51             if (st[i].mark[j]==2) k2++;

```

```

52         }
53         if ((k3==0) and (k2==0) and (k5<10)) {
54             cout<<i+1<<". " <<st[i].surname<<" " <<st[i].name<<
                    endl;
55
56             k=k+1;
57         }
58
59
60     }
61     cout<<"Number of good pupil = " <<k<<endl;
62
63
64 }
65
66 void Teacher::find_lagging_pupil(student *st)
67 {
68     int i;
69     int k=0;
70     for (i=1; i<16; i++){
71         int j;
72         int k5=0, k4=0;
73         for (j=0; j<10; j++){
74             if (st[i].mark[j]==5) k5++;
75             if (st[i].mark[j]==4) k4++;
76         }
77         if ((k5==0) and (k4==0)) {
78             cout<<i+1<<". " <<st[i].surname<<" " <<st[i].name<<
                    endl;
79
80             k=k+1;
81         }
82     }
83     cout<<"Number of lagging pupil = " <<k<<endl;
84 }
85
86 void Teacher::performance_calculation(student *st)
87 {
88     int i;
89     int k2=0, k4=0, k5=0, summ=0;
90
91     for (i=1; i<16; i++){
92         int j;
93         for (j=0; j<10; j++){
94             summ=summ+st[i].mark[j];
95             if (st[i].mark[j]==2) k2++;
96             if (st[i].mark[j]==4) k4++;
97             if (st[i].mark[j]==5) k5++;
98         }}

```



```

99     st[1].Quality=(100*(k4+k5))/150;
100    st[1].Progress=(150-k2)*10;
101    st[1].GPA=summ/150;
102 }
103
104 void Teacher::change_mark(student *st)
105 {
106     Pupil mPupil;
107     char TeacherSurname[20];
108     char TeacherName[20];
109     char password[20];
110
111     cout<<"Attention! Re-authentication! "<<endl;
112     cout<<"Enter your surname and name "<<endl;
113     cin>>TeacherSurname>>TeacherName;
114
115     cout<<"Enter password"<<endl;
116     cin>>password;
117
118     ifstream TeacherBook;
119     TeacherBook.open("../teacher.txt");
120
121     int i,error=0;
122     char s1[20];
123     char s2[20];
124     char PassChek[20];
125
126     for (i=1; i<10; i++){
127         TeacherBook>>s1>>s2>>PassChek;
128         if ((strcmp(s1,TeacherSurname)==0) and(strcmp(s2,
129             TeacherName)==0) and (strcmp(password,PassChek)
130                 ==0)){
131
132             int lenght_pass = strlen(password);
133             int new_mark;
134             cout<<"Enter student surname and name "<<endl;
135             cin>>mPupil.PupilSurname>>mPupil.PupilName;
136             cout<<"Enter new mark"<<endl;
137             cin>>new_mark;
138             for (int i=0; i<16; i++){
139                 if ((strcmp(st[i].surname,mPupil.PupilSurname)
140                     ==0) and(strcmp(st[i].name,mPupil.
141                         PupilName)==0)){
142                     st[i].mark[lenght_pass-1]=new_mark;
143                 }
144             }
145         }
146     }
147 }

```

```

144     if (error==1) cerr<<"Incorrect login or password"<<endl;
145     TeacherBook.close();
146     for (i=1; i<16; i++){
147         if ((strcmp(st[i].surname,mPupil.PupilSurname)==0)
148             and(strcmp(st[i].name,mPupil.PupilName)==0)){
149             cout<<st[i].mark[russian]<<" Russian"<<endl;
150             cout<<st[i].mark[literature]<<" Literature"<<endl;
151             ;
152             cout<<st[i].mark[algebra]<<" Algebra"<<endl;
153             cout<<st[i].mark[geometry]<<" Geometry"<<endl;
154             cout<<st[i].mark[history]<<" History"<<endl;
155             cout<<st[i].mark[SocialStudies]<<" Social Studies
156                 "<<endl;
157             cout<<st[i].mark[LSF]<<" Life Safety Fundamentals
158                 "<<endl;
159             cout<<st[i].mark[PE]<<" PE"<<endl;
160             cout<<st[i].mark[biology]<<" Biology"<<endl;
161             cout<<st[i].mark[geography]<<" Geography"<<endl;
162         }
163     }
164 }
165
166 Teacher::~Teacher()
167 {
168 }

```