

پروژه پیشرفته
مدیریت مهد کودک

نویسنده گزارشکار
اسماء چگنی

شماره دانشجویی
۹۹۱۲۳۵۸۰۱۵

در ادامه فاز اول پروژه کلاس Techer اضافه شد و به خاطر همین تغییراتی در کلاس login به وجود آمد چک کردن اطلاعات ن به روز شد تا بتواند کاربر معلم را نیز تشخیص دهد.

```
state Login::searchuser(string username, string password)
{
    ifstream Data("./userData.txt", ios::in); //open file
    if (Data.is_open())
    {
        while (1)
        {
            //read data from file
            Data >> setw(20) >> user.firstName >>
                setw(20) >> user.lastName >>
                setw(4) >> user.age >>
                setw(15) >> user.username >>
                setw(15) >> user.password;
            if (Data.eof())
            {
                break;
            }
            else
            {
                if (user.username == username && user.password == password) //check the password and username
                {
                    if (user.username[0] == 'M' && user.username[1] == 'A') //Check the first two letters of the username
                    {
                        return MANAGER; //user is manager
                    }
                    else if (user.username[0] == 'C' && user.username[1] == 'H')
                    {
                        return CHILD; //user is child
                    }
                    else if (user.username[0] == 'T' && user.username[1] == 'E')
                    {
                        return TEACHER; //user is teacher
                    }
                }
            }
        }
        return UNDEFINED_USER; //user undefined user
    }
}
```

توابع کلاس Mnagekids نیز نامشان تغییر کردن و برای چک کردن اطلاعات معلم نیز به روز شد

```
bool ManageKids::searchUser(string username)
{
    ifstream Data("./userData.txt", ios::in); //open file
    if (Data.is_open())
    {
        Data.clear(); //to read the correct information from the file
        while (1)
        {
            //read data from file
            Data >> setw(20) >> user.firstName >>
                setw(20) >> user.lastName >>
                setw(4) >> user.age >>
                setw(15) >> user.username >>
                setw(15) >> user.password;
            if (Data.eof())
            {
                break;
            }
            else
            {
                if (user.username == username)
                {
                    if (user.username[0] == 'C' && user.username[1] == 'H')
                    {
                        cout << "user founded" << endl;
                        Data.close();
                        return true;
                    }
                    else if (user.username[0] == 'T' && user.username[1] == 'E')
                    {
                        cout << "user founded" << endl;
                        Data.close();
                        return true;
                    }
                }
            }
        }
        Data.close();
        return false;
    }
}
```

کلاس Person تعریف شد که کلاس Manager و Teacher و Child از آن ارث میبرن در این کلاس توابع ست وگت مربوط به نام و نام خانوادگی تعریف شده است همچنین تابع setHomework در این کلاس تعریف شده است البته بدنه این تابع در این کلاس خالی و در کلاس Manager نیز کاربردی ندارد.

```
#ifndef PERSON_HPP
#define PERSON_HPP
#include <string>
class Person
{
public:
    Person();
    virtual void setFirstName(std::string);
    virtual void setLastName(std::string);
    virtual std::string getFirstName() const;
    virtual std::string getLastName() const;
    virtual void setHomework();
private:
    std::string FirstName;
    std::string LastName;
};
#endif //PERSON_HPP
```

کلاس های گفته شده به صورت public از این کلاس ارث برده اند.
در کلاس Teacher تابعی برای قرار دادن تکلیف بچه ها قرار داده شده است .
پیاده سازی توابع هنوز کامل و همه ی توابع مورد نیاز تعریف نشده اند در فاز دوم.
کلاس Teacher وکتوری از بچه ها را در خود دارد که با استفاده از این تابع برای
تمامی آنها تکلیف قرارداده میشود.

```
void Teacher::setHomework()
{
    int num1,num2,answer;
    cin>>num1>>num2>>answer;//get number for homework from teacher

    for (int i = 0; i < child.size(); ++i)
    {
        child[i].setHomework(num1 ,num2,answer);
    }
}
```

همین تابع نیز با پیاده سازی دیگری در کلاس Child قرار درد.

```
//  
void Child::setHomework(int num1 ,int num2 ,int answer)  
{  
    homework =new Homework(num1 ,num2,answer);  
}
```

در این کلاس اشاره گری از کلاس Homework تعریف شده است که زمانی که معلم تکلیفی را قرار میدهد در آن یک شی از کلاس Homework نیو میشود.
تابع doHomework اطلاعات شی homework را نشان میدهد تا بچه بتواند تکلیف را انجام دهد.
تابع دیگری هم برای چک کردن جواب بچه تعریف شده است که اگر جواب صحیح بود ۲۰ امتیاز به او اضافه میکند.
برای این کار هم متغیری برای ذخیره کردن امتیاز تعریف شده است.

```
void Child::doHomework()  
{  
    cout<<homework->getNum1()<<" + "<<homework->getNum2()<<" = ";  
    int result;  
    cin>>result;  
    homework->setResult(result);  
}  
//-----  
//-----  
void Child::setScore(int result)  
{  
    if(result==homework->getAnswer())  
    {  
        score+=20;  
    }  
}  
//-----  
//-----
```

کلاس Homework

این کلاس برای تکلیف بچه تعریف شده است

تکلیف بدین صورت است که دو عدد نمایش داده می شود و بچه باید حاصل جمع این دو عدد را حساب کرده و وارد کند.

در این کلاس دو عدد موردنظر باید کوچکتر از ۲۰ باشند

پاسخ صحیح هم در همین کلاس ذخیره میشود تا پس از انجام تکلیف توسط بچه پاسخ او چک شود و در صورت درست بودن به او امتیاز داده شود.

اینم مقادیر توسط معلم وارد میشوند .

```
1  #ifndef HOMEWORK_HPP
2  #define HOMEWORK_HPP
3  class Homework
4  {
5  public:
6      Homework(int ,int ,int);
7      void setNums(int ,int);
8      int getNum1()const;
9      int getNum2()const;
10     void setResult(int);
11     int getResult()const;
12     void setAnswer(int);
13     int getAnswer()const;
14 private:
15     int num1;
16     int num2;
17     int answer;
18     int result;
19 };
20 #endif //HOMEWORK_HPP
```

```
//-----  
int Homework::getNum1()const  
{  
    return this->num1;  
}  
//-----  
//-----  
int Homework::getResult()const  
{  
    return this->result;  
}  
//-----  
//-----  
int Homework::getNum2()const  
{  
    return this->num2;  
}  
//-----  
//-----  
void Homework::setAnswer(int answer)  
{  
    if(answer==num1+num2)  
    {  
        this->answer=answer;  
    }  
}  
//-----  
//-----  
int Homework::getAnswer()const  
{  
    return this->answer;  
}  
//-----  
//-----  
void Homework::setResult(int result)  
{  
    this->result=result;  
}
```