

پروژه پیشرفته  
مدیریت مهد کودک

نویسنده گزارشکار  
اسماء چگنی

شماره دانشجویی  
۹۹۱۲۳۵۸۰۱۵

در فاز سوم اکپشن ها به کلاس ها اضافه شد و در همان جا در بلاک try و catch قرار داده شد تا اجرای برنامه متوقف نشود.

ایجاد تکلیف توسط معلم با استفاده از گرفتن سه عدد از وردی و نوشتن آنها در فایل صورت میگیرد.

به همین ترتیب برای ست کردن تکلیف در کلاس بچه تکلیف از همان فایل خوانده شده و اعداد در کلاس تکلیف قرار میگیرند.

```
void Teacher::writeToFile(int num1 ,int num2 ,int answer)
{
    ofstream Home("./Homework.txt", ios::out); //open file
    if (Home.is_open()) {
        Home.clear();
        //write new data in file
        Home << left << setw(۳) << num1 << setw(۳) << num2 << setw(۳) <<
answer << endl;
        Home.close();
    }
    else
    {
        cerr << "Can not open the file!" << endl;
    }
}
```

```
void Teacher::setHomework()
{
    try {
        cout<<"Enter num1 ,num2 ,answer : ";
        int num1, num2, answer;
        cin >> num1 >> num2 >> answer; //get number for homework from teacher
        if (num1 < ۲۰ && num2 < ۲۰ && ((num1 + num2) == answer)) {
            writeToFile(num1,num2,answer);
            cout<<" set homework "<<endl;
        } else {
            throw invalid_argument("Data is not valid");
        }
    }
    catch (invalid_argument &e)
    {
        cout<<e.what()<<endl;
    }
}
```

```

void Child::readFromFile()
{
    ifstream Home("./Homework.txt", ios::in); //open file
    if (Home.is_open()) {
        int num1, num2, answer;
        Home.clear();
        //write new data in file
        Home >> setw(۳) >> num1 >> setw(۳) >> num2 >> setw(۳) >> answer ;
        cout<<num1<<endl;
        cout<<num2<<endl;
        cout<<answer<<endl;
        setHomework(num1, num2, answer);
        Home.close();
    }
    else
    {
        cerr << "Can not open the file!" << endl;
    }
}

```

دو کلاس square و matchgame نیز در این فاز تعریف شده اند

که یک بازی ساده است بدین صورت که بازی از یک جدول با ۱۲ خانه تشکیل شده است و در هر مرحله بچه می تواند دو خانه را انتخاب کند در صورتی که تصویر ئو خانه یکسان باشد به او ده امتیاز تعلق میگرد.

کلاس square هرکدام از خانه های جدول است که یک ویژگی آدرس عکس و مچ شدن یا نشدن را دارند.

اپریتورهای == و! نیز برای این کلاس پیاده سازی شده است.

== این اپریتور آدرس عکس های دو خانه انتخاب شده را باهم مقایسه می کند.

! مچ شدن یا نشدن خانه را برمیگرداند.

که در صورتی که مچ شده باشد پیغام خطا نشان داده میشود.

در کلاس matchgame یک وکتور دوبعدی از pair قرار دارد که از اینت و کلاس square تشکیل شده است.

با انتخاب دو عدد آدرس عکس های دوخانه و مچ شدن یا نشدن آن دوخانه چک میشود.

در کلاس child یک اشاره گر از این کلاس بازی قرار داده شده است که اگر تابع playing فراخوانی شود تا ۶ مرحله میتواند این بازی را انجام دهد.

```
class Square
{
public:
    Square(std::string);
    void setUrl(std::string);
    std::string getUrl() const;
    void setMatch(bool);
    bool getMatch() const;
    bool operator==(const Square &);
    bool operator!();
private:
    std::string url="";
    bool isMatch= false;
};
```

```
void MatchGame::checkUrl(int square۱ ,int square۲)
{
    pair <Square,int>Square۱= searchSquare(square۱);
    pair <Square,int>Square۲= searchSquare(square۲);
    if(!Square۱.first && !Square۲.first) {
        if (Square۱.first == Square۲.first) {
            cout << "You guessed it right" << endl;
            setIsMatch(square۱,square۲);

            setScore();
            cout<<getScore()<<endl;
        } else {
            cout << "You guessed wrong" << endl;
        }
    } else
    {
        throw invalid_argument("The houses are empty");
    }
}
```

```
void Child::playing()
{
    cout<<"Game Started!"<<endl;
    int num\,numʹ;
    matchGame=new MatchGame;
    matchGame->setSquares();
    for (int i = 0; i < 6; ++i)
    {
        cout<<"Enter num\ AND numʹ :  ";
        cin>>num\>>numʹ;
        matchGame->checkUrl (num\,numʹ);
    }
}
```