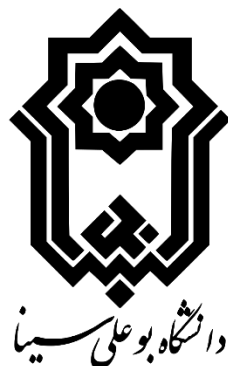


بسمه تعالی



عنوان:

«گزارش کار فاز اول پروژه پایانی درس برنامه نویسی پیشرفته»

«Sewing Game»

استاد:

سرکار خانم مهندس بطحائیان

دانشجو:

نسترن منصوری

۹۶۱۲۳۵۸۰۴۱

ترم تحصیلی:

۴۰۰۱

فهرست

۳	مقدمه:.....
۳	توضیحات کلی پروژه نهایی:
۴	کلاس های اولیه:.....
۴	کلاس Tailor:.....
۶	کلاس Color:.....
۸	کلاس Design:.....
۱۰	تابع main():.....

مقدمه:

در این فاز هدف، پیاده سازی تعدادی از کلاس های اولیه پروژه نهایی است.

توضیحات کلی پروژه نهایی:

پروژه ی نهایی پیاده سازی بازی خیاطی است. در این بازی کاربر به عنوان خیاط وارد بازی می شود و لباس های زیبا طراحی میکند. به این صورت که هنگام ورود نام خود را به عنوان خیاط می تواند وارد کند. سپس در قسمت بعدی پارچه ها با رنگ های متفاوت به کاربر نشان داده می شوند . کاربر میتواند یکی از آن ها را انتخاب کند. بعد از انتخاب رنگ پارچه، کاربر باید در مرحله بعد طراحی موردنظر خود برای پارچه را از بین طراحی های موجود انتخاب می کند. در ادامه پیاده سازی کلاس های اولیه برای این بازی توضیح داده می شود.

کلاس‌های اولیه:

کلاس Tailor:

این کلاس، کلاس خیاط است؛ که در دو قسمت header و .cpp تعریف شده است. در این کلاس

یک متغیر خصوصی تحت نام `tailor_name` برای ذخیره نام کاربر و دو تابع `set_tailor_name` و

`get_tailor_name` وجود دارد که این توابع برای ذخیره کردن و برگرداندن نام در متغیر خصوصی

`tailor_name` هستند.

```
#include <iostream>
#include <string>

#ifndef TAILOR_H
#define TAILOR_H
//class definition
class Tailor
{
public:

    Tailor(std::string ="unknown"); //default constructor

    void set_tailor_name(std::string &); //set tailor name
    std::string get_tailor_name() const; //return tailor name

private:
    std::string tailor_name; //for save the tailor or user name
};

#endif // TAILOR_H
```

Figure 1: پیاده سازی header در کلاس Tailor

همانطور که در تصویر بالا مشخص است در قسمت کانستراکتور ورودی یک متغیر از نوع `string` می

باشد که دارای مقدار پیش فرض "`unknown`" است؛ به این معنی که به محض ورود کاربر به محیط بازی نام

اون `unknown` خواهد بود و کاربر میتواند آن را تعویض کند و نام مورد علاقه خود را به جای آن بگذارد.

```

#include <iostream>
#include <string>
using namespace std;

Tailor::Tailor(string name) //Implement a constructor that replaces the
{                           //default value of the unknown tailor name.
    set_tailor_name(name);
}

void Tailor::set_tailor_name(string & name) //This function saves it in variable tailor_name
{                                           //after the user enters the tailor's name.
    tailor_name=name;
}

string Tailor::get_tailor_name() const    | //This function returns the tailor name.
{
    return tailor_name;
}

```

Figure ۲: پیاده سازی .cpp کلاس Tailor

در پیاده سازی بدنه کلاس Tailor در قسمت اول پیاده سازی کانستراکتور وجود دارد؛ که در این

قسمت بعد کانستراکتور بعد از دریافت ورودی از نوع رشته ، تابع set_tailor_name فراخوانی می کند.

پیاده سازی تابع set_tailor_name: این تابع بعد از دریافت ورودی از نوع رشته ورودی را در متغیر

tailor_name ذخیره می کند.

پیاده سازی تابع get_tailor_name: این تابع مقدار موجود در tailor_name را برمیگرداند.

کلاس Color:

این کلاس، کلاس رنگ است و برای تعیین رنگ پارچه توسط کاربر تعریف شده است؛ که در دو قسمت

header و cpp. تعریف شده است. در این کلاس یک متغیر خصوصی تحت نام fab_color برای ذخیره

رنگ پارچه و دو تابع set_color و get_color وجود دارد که این توابع برای ذخیره کردن و برگرداندن رنگ

پارچه در متغیر خصوصی fab_color هستند.

```
#include<iostream>
#include <string>

#ifndef COLOR_H
#define COLOR_H

//class definition
class Color
{
public:
    Color(std::string = "white"); //default constructor
    void set_color(std::string); //set color for fabric
    std::string get_color() const; //return color of fabric

private:
    std::string fab_color;
};

#endif // COLOR_H
```

Figure ۳: پیاده سازی header کلاس Color

در header این کلاس مانند کلاس قبل یک کانستراکتور با مقدار پیش فرض وجود دارد که مقدار پیش فرض

رنگ پارچه را White یا همان سفید قرار می دهد.

```

#include <iostream>
#include <string>

#include "Color.h"

using namespace std;

Color::Color(string col)    //Sets white by default.
{
    set_color(col);
}

void Color::set_color(string col)    //This function saves it in variable fab_color.
{
    fab_color=col;
}

string Color::get_color() const    //This function returns the fabric color.
{
    return fab_color;
}

```

Figure ۴: پیاده سازی .cpp کلاس Color

در پیاده سازی بدنه کلاس Color در قسمت اول پیاده سازی کانستراکتور وجود دارد؛ که در این قسمت کانستراکتور بعد از دریافت ورودی از نوع رشته ، تابع set_color فراخوانی می کند.

پیاده سازی تابع set_color: این تابع بعد از دریافت ورودی از نوع رشته ورودی را در متغیر fab_color ذخیره می کند.

پیاده سازی تابع get_color: این تابع مقدار موجود در fab_color را برمیگرداند.

کلاس Design:

این کلاس، کلاس طراحی است و برای انتخاب نوع طراحی برای پارچه توسط کاربر پیاده سازی شده است؛ که در دو قسمت header و cpp. تعریف شده است. در این کلاس یک متغیر خصوصی تحت نام type_des برای ذخیره نوع طراحی وجود دارد. همچنین سه تابع دیگر هم در این کلاس وجود دارند که در ادامه توضیح داده می شوند.

```
#include <iostream>
#include <string>

#ifndef DESIGN_H
#define DESIGN_H

class Design
{
public:
    Design(std::string = "empty");
    void choose_design();
    void set_design(std::string);
    std::string get_design() const;

private:
    std::string type_des;
};

#endif // DESIGN_H
```

Figure ۵: پیاده سازی header کلاس Design

در header این کلاس مانند دو کلاس قبل یک کانستراکتور با مقدار پیش فرض وجود دارد که مقدار پیش فرض نوع طراحی لباس را empty به معنای خالی قرار می دهد و تا کاربر طراحی مورد نظر خود را انتخاب نکند نمی تواند به مرحله ی بعد برود.


```

#include "Design.h"
using namespace std;

Design::Design(string des)
{
    set_design(des);
}

void Design::choose_design()
{
    cout<<"Choose the design you want:";
    cout<<"\t 1.Mantue\t2.Pans\t 3.scarf:\t";
    cout<<endl;
}

void Design::set_design(string des)
{
    type_des=des;
}

string Design::get_design() const
{
    return type_des;
}

```

Figure ۶: پیاده سازی کلاس Design.cpp

در پیاده سازی بدنه کلاس Design در قسمت اول پیاده سازی کانستراکتور وجود دارد؛ که در این

قسمت کانستراکتور بعد از دریافت ورودی از نوع رشته ، تابع set_design را فراخوانی می کند.

پیاده سازی تابع choose_design: این تابع بعد از فراخوانی از کاربر میخواهد تا طراحی موردنظر

خود را از لیستی که به اون می دهد انتخاب کند. همانطور که در تصویر ۶ مشخص است این لیست شامل موارد زیر است:

- Mantue
- Pans
- Scarf

پیاده سازی تابع set_design: این تابع طراحی انتخاب شده توسط کاربر را در متغیر type_des

ذخیره می کند.

پیاده سازی تابع get_design: این تابع مقدار ذخیره شده در type_des که همان طرح انتخاب

شده توسط کاربر می باشد را بر می گرداند.

تابع main():

در قسمت اول تابع main یک شیء از کلاس Tailor ایجاد می شود سپس مقدار پیش فرض آن نشان داده شده و از کاربر خواسته می شود تا نام خود را وارد کند. سپس نام وارد شده توسط کاربر را برمیگرداند تا از صحت آن مطمئن شوید.

```
int main()
{
    string st1,st2,st3;           //st1=save tailor name //st2=save fabric color //st3=save type of design
    // cout << "Hello world!" << endl;
    Tailor t1;
    cout<<"Default tailor name:\t"<<t1.get_tailor_name()<<endl<<endl;    //check default name

    cout<<"Enter your tailor name:\t";
    getline(cin,st1);

    t1.set_tailor_name(st1);
    cout<<"Set your tailor name:\t"<<t1.get_tailor_name()<<endl<<endl;
    //*****//
}
```

Figure ۷: پیاده سازی قسمت اول تابع main

در قسمت دوم تابع main یک شیء از کلاس Color ایجاد می شود. سپس مقدار پیش فرض آن به کاربر نمایش داده می شود و از کاربر خواسته میشود تا از لیست موجود یک رنگ را به دلخواه انتخاب کرده و آن را وارد کند. سپس رنگ ورودی توسط کاربر نمایش داده می شود.

```
Color c1;

cout<<"Default color is\t"<<c1.get_color()<<endl<<endl;    //check default color

cout<<"Choose your favorite color from the list below:"<<endl<<endl;
cout<<"Blue, Yellow, Green, Black, Red"<<endl<<endl;

cout<<"Enter your favorite color:\t";
cin>>st2;

c1.set_color(st2);
cout<<"Set fabric color:\t"<<c1.get_color()<<endl<<endl;
```

Figure ۸: پیاده سازی قسمت دوم تابع main

در قسمت سوم تابع main هم یک شیء از کلاس Design ایجاد شده است. بعد مقدار پیش فرض ذخیره شده که empty است نمایش داده می شود و در نهایت از کاربر خواسته می شود که طراحی مورد علاقه خود را انتخاب کند و بعد مقدار تعیین شده نمایش داده می شود.