



برنامه نویسی پیشرفته – ترم 4001

فاز اول پروژه شماره 4 – مکانیک اتومبیل

استاد : مهندس بطحائیان

محمدحسین الطافی - 9912358004

مقدمه

این پروژه درباره مکانیکی است که ابزارهای مورد نیاز خودش را تهیه می کند.

شرح فاز اول

در فاز اول , دو کلاس "پیچگوشتی" و "چراغ ماشین" پیاده سازی شده و در صورت نیاز در فاز های بعدی کلاس های بیشتر نیز اضافه می شوند. هر کلاس چند private member و چند تابع set و get و یک تابع کانستراکتور دارد.

کلاس مکانیک برای مقدار پول و... در فاز های بعد اضافه می شود.

ساختار برنامه

کلاس screwdriver (پیچگوشتی) و توابع و enum های مربوط به آن:

```
enum screwdriver_head_size {small_head = 1 , medium_head , large_head};
enum screwdriver_head_type {slothead = 1 , phillipshead , pozidriv , torx , allen};
enum screwdriver_length {extrasmall_len = 1 , small_len , medium_len , large_len , extralarge_len};

class screwdriver
{
public:
    screwdriver(screwdriver_head_size headsize , screwdriver_head_type headtype , screwdriver_length len , float prc);
    void set_head_size(screwdriver_head_size headsize);
    screwdriver_head_size get_head_size() const;
    void set_head_type(screwdriver_head_type headtype);
    screwdriver_head_type get_head_type() const;
    void set_length(screwdriver_length len);
    screwdriver_length get_length() const;
    void set_price(float prc);
    float get_price() const;
    void printinfo() const;

private:
    screwdriver_head_size head_size;
    screwdriver_head_type head_type;
    screwdriver_length length;
    float price;
};
```

برای سایز سر پیچگوشتی enum screwdriver_head_size در نظر گرفته شده که 3 حالت دارد.

برای نوع پیچگوشتی enum screwdriver_head_type در نظر گرفته شده که 5 حالت دارد.

برای طول پیچگوشتی enum screwdriver_length در نظر گرفته شده که 5 حالت دارد.

از این enum ها به عنوان نوع داده های private member ها استفاده شده است و یک private member برای قیمت نیز وجود دارد.

توابع set و get برای هر کدام از 4 پرایوت ممبر نوشته شده که متن بدنه آنها در screwdriver.cpp است.

تابع کانستراکتور با فراخوانی توابع set مقادیر را در پرایوت ممبر ها قرار می دهد. در توابع set صحت ورودی ها چک می شود و در صورت اشتباه بودن, اکسپشن پرتاب می شود و در تابع کانستراکتور اکسپشن دریافت می شود و حالت پیشفرض, جایگزین مقدار اشتباه می شود.

تابع printf برای چاپ اطلاعات استفاده می شود. (از enum ها در switch case مربوط به این تابع نیز استفاده می شود.)

کلاس carlight (چراغ ماشین) و توابع و enum های مربوط به آن :

```
enum lighttype {headlight = 1 , fog_light , blinker , brake_light};
enum color {white = 1 , yellow , red , blue};
enum brightness {low = 1 , mid , high};

class carlight
{
public:
    carlight(color col , lighttype type , brightness brght , float prc);
    void set_color(color col);
    color get_color() const;
    void set_light_type(lighttype type);
    lighttype get_light_type() const;
    void set_brightness(brightness brght);
    brightness get_brighrness() const;
    void set_price(float prc);
    float get_price() const;
    void printinfo() const;

private:
    color chosencolor;
    lighttype light_type;
    brightness chosenbrightness;
    float price;
};
```

برای نوع چراغ, enum lighttype در نظر گرفته شده که 4 حالت دارد.
برای رنگ چراغ, enum color در نظر گرفته شده که 4 حالت دارد.
برای شدت روشنایی چراغ, enum brightness در نظر گرفته شده که 3 حالت دارد.

از این enum ها به عنوان نوع داده های private member ها استفاده شده است و یک private member برای قیمت نیز وجود دارد.
توابع set و get برای هر کدام از 4 پرایوت ممبر نوشته شده که متن بدنه آنها در carlight.cpp است.

تابع کانستراکتور با فراخوانی توابع set مقادیر را در پرایوت ممبر ها قرار می دهد. در توابع set صحت ورودی ها چک می شود و در صورت اشتباه بودن, اکسپشن پرتاب می شود و در تابع کانستراکتور اکسپشن دریافت می شود و حالت پیشفرض, جایگزین مقدار اشتباه می شود.

تابع printfinfo برای چاپ اطلاعات استفاده می شود. (از enum ها در switch case مربوط به این تابع نیز استفاده می شود).

تابع main :

در تابع main , چند نمونه فراخوانی کانستراکتور برای تست کردن برنامه وجود دارد که با ورودی های درست یا نادرست, کانستراکتور ها فراخوانی می شوند.

در صورت اشتباه بودن, پیام ارور مربوط به آن و مقدار پیشفرضی که قرار می گیرد برای کاربر چاپ می شوند.

محمدحسین الطافی

9912358004