

برنامه سازی پیشرفته

گزارشکار فاز اول پروژه

هانیه اکبری

شماره دانشجویی : ۹۹۱۲۳۵۸۰۰۳

Carpenter.h :

```
enum WoodType
{
    cherrywood,           //choobe gilas
    walsawood,            //choob walsa
    blackmaple,           //afraze sia
    walnut                //choob gerdo
};
```

در ابتدا یک متغیر از نوع enum برای جنس‌های مختلف چوب تعریف کردم.

```
class Wood
{
public :
    Wood (WoodType woodtype) : type (woodtype)           //constructor
    {
    }
    bool set_length (float l);
    bool set_width (float w);
    bool set_height (float h);
    float get_length ();
    float get_width ();
    float get_height ();

private :
    float length;
    float width;
    float height;
    const WoodType type;
};
```

کلاس wood برای چوب تعریف شده به این صورت که ۴ متغیر برای ابعاد و جنس چوب در نظر گرفته شده. جنس چوب یک متغیر ثابت است زیرا تغییری در آن رخ نمی‌دهد. این متغیرها در قسمت private تعریف شده تا سطح دسترسی به آن‌ها برای همه وجود نداشته باشد و توابع set و get و constructor در قسمت public کلاس تعریف شدند.

تابع constructor یک تابع سازنده هم نام با کلاس است که وقتی یک شی از کلاس تعریف می‌شود constructor قبل از اینکه داده‌ها فراخوانی شوند، فراخوانی می‌شود.

(وقتی ما متغیر const داریم باید حتما اون متغیر و مقداردهی کنیم)

```

class Tool
{
    public :
        string set_name(string n);
        bool set_type (int t);
        bool set_size (int s);
        int get_size ();
        string get_name ();
        int get_type ();

    private :
        string name_tool;
        int type_tool;                //electric or non-electric
        (0 for non-electric, 1 for electric)
        int size_tool;                //3 for big, 2 for medium,
        1 for little
};

```

در کلاس ابزار یک متغیر برای نام ابزار، یک متغیر برای سایز ابزار که می‌تواند بزرگ، کوچک یا متوسط باشد و یک متغیر برای نوع ابزار، الکتریکی یا غیرالکتریکی بودن آن.

توابع set و get آن‌ها هم در قسمت public کلاس تعریف شدند.

Carpenter.cpp:

```

bool Wood :: set_length (float l)
{
    if (l > 0)                //Check the length of the
    positive
    {
        length = l;
        return true;
    }
    else
        return false;
}

```

تابع set_length یک پارامتر را به عنوان ورودی می‌گیرد و اگر آن عدد مثبت بود آن را داخل متغیر length می‌ریزد. در غیر این صورت false را برمی‌گرداند. برای عرض و ارتفاع چوب هم چنین توابعی تعریف شدند.

```
float Wood :: get_length ()
{
    return length;
}
```

تابع `get_length` مقدار `length` را برمی‌گرداند. برای سایر ابعاد چوب نیز چنین تابعی تعریف شده‌اند.

```
string Tool :: set_name(string n)
{
    name_tool = n;
    return n;
}
```

تابع `set_name` یک متغیر از نوع `string` می‌گیرد و آن را داخل متغیر `name` می‌ریزد. این تابع مربوط به کلاس ابزار است.

```
bool Tool :: set_type (int t)
{
    if (t == 0 || t == 1) //The number
entered must be 0 or 1
    {
        type_tool = t;
        return true;
    }
    else
        return false;
}
```

تابع `set_type` با دریافت یک پارامتر از نوع `int` ابتدا چک می‌کند که این عدد حتما صفر (برای غیرالکتریکی) یا یک (برای الکتریکی) باشد سپس داخل متغیر `type_tool` می‌ریزد.

```
bool Tool :: set_size (int s)
{
    if (s == 1 || s == 2 || s == 3) //The number
entered must be 1, 2, or 3
    {
        size_tool = s;
        return true;
    }
    else
        return false;
}
```

تابع `set_size` با دریافت یک پارامتر از نوع `int` ابتدا چک می‌کند عدد دریافتی حتما ۱ یا ۲ یا ۳ باشد، ۱ برای سایز کوچک ۲ برای سایز متوسط و ۳ برای سایز بزرگ و بعد آن را در متغیر `size_tool` می‌ریزد.

```
string Tool :: get_name()
{
    return name_tool;
}
```

تابع `get_name` نام ابزار را برمی گرداند. برای سائز و نوع هم این تابع تعریف شده است.