类与类的封装

> C++通过`public`、`protected`、`private`三个关键字来控制成员变量和成员函数的访问权限，被称为`成员访问限定符`，成员访问限定符与类的封装有关。
>
> > C++中的成员访问限定符只能修饰类里面的成员，不能修饰类
> >
> > 1. 在类的内部（定义类的代码内部），无论成员被声明为`public`、`protected`还是`private`，都是可以互相访问的，没有访问权限的限制。
> > 2. 在类的外部（定义类的代码之外），**只能通过对象访问成员，并且通过对象只能访问public属性的成员**，不能访问`private`、`protected`属性的成员。

`lesson_15-1.cpp`

```cpp
#include<stdio.h>
struct Biology {
    bool living;
};
struct Animal : Biology {
    bool movable;
    void findFood(){};
};
struct Plant : Biology {
    bool growable;
};
struct Beast : Animal {
    void sleep(){};
};
struct Human : Animal {
    void sleep(){};
    void work(){};
};

struct Girl : Human
{
private:
    int age;
    int weight;
public:
    void print()
    {
        age = 22;
        weight = 44;
        printf("girl age = %d\tweight = %d\n", age, weight);
    }
};

struct Boy : Human
```

```
{
private:
    int height;
    int salary;

public:
    int age;
    int weight;
    void print()
    {
        height = 177;
        salary = 900;
        printf("boy height = %d\t salary = %d\n", height, salary);
    }


};


int main()
{
    Girl g;
    Boy b;

    // printf("g.age = %d\n", g.age);  // error: ‘int Girl::age’ is private within this context
    g.print();

    // b.height = 100;  // error: ‘int Boy::height’ is private within this context

    printf("b.age = %d\nb.weight = %d\n", b.age, b.weight);
    return 0;
}
```

> `g.age`：这是一个私有成员变量，无法通过对象访问`private`修饰的成员变量，`b.height`也是

类成员的作用域

- 类成员的作用域都只在类的内部，外部无法直接访问
- 成员函数可以**直接访问**成员变量和调用成员函数，即使目标成员变量和目标成员函数被`private`或者`protected`修饰也是可以被成员函数直接访问的
- 类的外部可以**通过类变量(对象)**访问`public`修饰的成员
- C++中用`struct`定义的类中所有成员默认为`public`

`lesson_15-2.cpp`

```
#include <stdio.h>

```

```cpp
int i = 1;
struct Test
{
private:
    int i;
public:
    int j;

    int getI()
    {
        i = 3;
        return i;
    }

};

int main(void)
{
    int i = 2;

    Test test;
    test.j = 4;

    printf("i = %d\n", i);  // 2
    printf("::i = %d\n", ::i);  // 1
    // printf("test.i = %d\n", test.i);  // random value
    printf("test.j = %d\n", test.j);  // 4
    printf("test.getI() = %d\n", test.getI());  // 3
}
```

> `::` 表示默认的命名空间，也就是全局作用域
> 想要访问一个类的成员变量或者成员函数，必须存在这个类的对象，通过这个类的对象访问是否成功，还需要看成员的访问权限

小结

- C++中可以通过成员访问限定符实现封装机制
- `public` 成员可以在类的内部和外部访问或调用
- `private` 成员只能在类的内部被访问和调用