

第五课：多文件项目和访问修饰符

多文件项目结构

```
Lesson05/
├─ Program.cs          // 入口，只管 Main
├─ Models/
│   ├─ Person.cs       // 数据模型
│   └─ Microphone.cs   // 麦克风类
├─ Helpers/
│   └─ MathHelper.cs   // 工具类
└─ Services/
    └─ AudioService.cs // 服务类
```

关键规则：

- 同一项目内，所有 `.cs` 文件自动互相可见（不需要 import）
- 用文件夹组织，但编译时所有文件合并成一个程序集

public vs private 核心区别

修饰符	类内访问	类外访问	继承访问
public	✓	✓	✓
private	✓	✗	✗

```
class Person
{
    public string name;      // 任何地方都能读写
    private int age;         // 只有 Person 内部能访问
    private string secret;   // 外部无法访问

    public void SetAge(int age)
    {
```

```

        // 公有方法内部可以访问私有字段
        if (age > 0 && age < 150)
            this.age = age;
    }

    public int GetAge()
    {
        return this.age; // 通过方法间接访问
    }
}

```

```

// Program.cs
Person p = new Person("无言势");
p.name = "新名字";           // OK, public
p.age = 25;                   // 编译错误, private
p.SetAge(25);                 // OK, 通过公有方法

```

为什么要 private?

数据保护：防止无效值

```

private int age;

// 外部不能直接 age = -100
// 必须通过方法验证后设置
public void SetAge(int age)
{
    if (age < 0) age = 0;
    if (age > 150) age = 150;
    this.age = age;
}

```

封装：内部实现可以随时改，不影响外部

```

// 内部存储从 int 改成 DateTime 都可以
// 只要 public 方法签名不变，外部代码不用改

```

其他访问修饰符

修饰符	说明
<code>public</code>	完全公开
<code>private</code>	仅本类
<code>protected</code>	本类 + 子类
<code>internal</code>	同项目内

属性（Property） - 介于 public/private 之间

```
class Person
{
    private int _age;

    public int Age
    {
        get { return _age; }           // 读取
        set
        {
            if (value < 0) value = 0;  // 写入时验证
            if (value > 150) value = 150;
            _age = value;
        }
    }
}
```

```
Person p = new Person();
p.Age = 25;           // 自动调用 set
int a = p.Age;        // 自动调用 get
```