**抽象类、密封类及类成员（C# 编程指南）**

使用 [abstract](https://msdn.microsoft.com/zh-CN/library/sf985hc5.aspx) 关键字可以创建不完整且必须在派生类中实现的类和[类](https://msdn.microsoft.com/zh-CN/library/0b0thckt.aspx)成员。

使用 [sealed](https://msdn.microsoft.com/zh-CN/library/88c54tsw.aspx) 关键字可以防止继承以前标记为 [virtual](https://msdn.microsoft.com/zh-CN/library/9fkccyh4.aspx) 的类或某些类成员。

**抽象类和类成员**

通过在类定义前面放置关键字 **abstract**，可以将类声明为抽象类。例如：

public abstract class A

{

// Class members here.

}

抽象类不能实例化。抽象类的用途是提供一个可供多个派生类共享的通用基类定义。例如，类库可以定义一个抽象类，将其用作多个类库函数的参数，并要求使用该库的程序员通过创建派生类来提供自己的类实现。

抽象类也可以定义抽象方法。方法是将关键字 **abstract** 添加到方法的返回类型的前面。例如：

public abstract class A

{

public abstract void DoWork(int i);

}

抽象方法没有实现，所以方法定义后面是分号，而不是常规的方法块。抽象类的派生类必须实现所有抽象方法。当抽象类从基类继承虚方法时，抽象类可以使用抽象方法重写该虚方法。例如：

// compile with: /target:library

public class D

{

public virtual void DoWork(int i)

{

// Original implementation.

}

}

public abstract class E : D

{

public abstract override void DoWork(int i);

}

public class F : E

{

public override void DoWork(int i)

{

// New implementation.

}

}

如果将 **virtual** 方法声明为 **abstract**，则该方法对于从抽象类继承的所有类而言仍然是虚方法。继承一个抽象方法的类不能访问该方法的原始实现。在上一个示例中，类 F 中的 DoWork 不能调用类 D 中的 DoWork。通过这种方式，抽象类可以强制派生类为虚方法提供新的方法实现。

**密封类和类成员**

通过在类定义前面放置关键字 **sealed**，可以将类声明为[密封](https://msdn.microsoft.com/zh-CN/library/88c54tsw.aspx)类。例如：

public sealed class D

{

// Class members here.

}

密封类不能用作基类。因此，它也不能是抽象类。密封类禁止派生。由于密封类从不用作基类，所以有些运行时优化可以略微提高密封类成员的调用速度。

在对基类的虚成员进行重写的派生类上，方法、索引器、属性或事件可以将该成员声明为密封成员。在用于以后的派生类时，这将取消成员的虚效果。方法是在类成员声明中将 **sealed** 关键字置于 [override](https://msdn.microsoft.com/zh-CN/library/ebca9ah3.aspx) 关键字的前面。例如：

public class D : C

{

public sealed override void DoWork() { }

}

## 请参阅

[C# 编程指南](https://msdn.microsoft.com/zh-CN/library/67ef8sbd.aspx)

[类和结构（C# 编程指南）](https://msdn.microsoft.com/zh-CN/library/ms173109.aspx)

[继承（C# 编程指南）](https://msdn.microsoft.com/zh-CN/library/ms173149.aspx)

[方法（C# 编程指南）](https://msdn.microsoft.com/zh-CN/library/ms173114.aspx)

[字段（C# 编程指南）](https://msdn.microsoft.com/zh-CN/library/ms173118.aspx)

[如何：定义抽象属性（C# 编程指南）](https://msdn.microsoft.com/zh-CN/library/yd3z1377.aspx)