

继承的概念

继承主要为了少些代码,子类可以复用父类的代码,可以理解为把父类中的代码复制到了子类中

继承的语法

```
1 class A
2 {
3 }
4 class B:A //表示B类继承自A类
5 {
6 }
```

Object万类之祖

- 1.Csharp中所有类默认继承自 Object
- 2.Csharp中任何数据类型都可以隐式转换成Object
- 3.引用类型转Object无额外性能开销
- 4.值类型转Object会产生装箱操作 反之有拆箱操作 两者都有性能开销

继承的优点

- 1.代码复用
- 2.便于代码的层次化管理
- 3.提高代码的可维护性和可扩展性

继承的缺点

- 1.耦合度比较高,父类的修改会影响到子类
- 2.层层继承可能导致子类臃肿,包含很多用不到的数据

重写

当父类中的方法在子类中不适用时,需要重写父类的方法

virtual关键字:虚

修饰方法表示方法可以被子类重写

override 关键字:重写

重写父类中的方法(多态性的表现)

只有父类中被 `virtual abstract override`关键字修饰的方法可以被重写

abstrac关键字:抽象

修饰方法表示抽象方法

抽象方法不包含方法体

抽象方法在子类中必须实现,除非子类也是抽象类

抽象方法必须包含在抽象类中

修饰类表示抽象类

抽象类可以包含普通类的成员,可以包含有实现的方法

抽象类不可以创建对象

父类引用指向子类对象

例如狗类继承自动物类那么可以这样写:

```
1  动物 a = new 狗();
2  动物[] arr={new 狗(),new 猫(),new 狗()}
```

父子类型2种转换方式:

- 1 子转父: 隐式转换 (自动转换)
- 2 父转子: 需要用到用到`as`转换 或者 `()` 强制转换
- 3 1.使用`as`关键字转换如果转换不成功返回`null`,只能用于引用类型
- 4 2使用 `()` 强制转换如果不成功则抛出异常

继承中构造函数调用过程

- 1 先调用父类中的构造函数再调用子类的构造函数
- 2 默认调用父类公共无参构造函数, 可以使用: `base` 调用指定的父类构造函数

使用new关键字覆盖父类成员(了解)

在子类中定义父类中已经存在的成员的时候, 用new关键字指明是覆盖父类成员
new和override的区别:

当父类引用指向子类对象得时候:

覆盖: 调用父类中的成员

重写: 永远调子类的成员

sealed关键字: 密封

修饰类 类不能被继承
修饰已经被重写过的方法, 则方法不能再被重写

面向对象语言三大特征:

- | | |
|------|-----------------------------|
| 1 封装 | 体现在访问修饰符 |
| 2 继承 | 复用父类的代码 |
| 3 多态 | 父类中定义的行为在子类中有不同的表现形式. 体现在重写 |

访问修饰符

public 公开的 哪儿都能访问

private 私有的 只能在本类内部访问

protected 受保护的 只能本类或子类内部可以访问

internal 内部的 只能在本程序集内部可以访问

protected internal 受保护的内部的 程序集内和Internal一样 程序及外和protected 一样

程序集:

两种形态: dll exe

创建程序集: 创建项目的时候选择类库, 写完代码点击生成即可生成dll文件

引用程序集:在解决方案管理器>项目>引用上右键>添加引用>浏览>选择dll文件,写代码的时候记得引用命名空间

面向对象设计七大原则

依赖倒置:依赖抽象

李氏替换:父类出现的地方都可以用子类替换

单一职责:自己的事情自己做,不要管别人的

接口隔离:尽量定义小而精的接口

迪米特法则:不要跟陌生说话,类与类尽量少交互

开闭原则:对扩展开放,对修改关闭

组合优先于继承:因为继承会给子类带来用不到的成员,浪费内存

作业:

抽象父类:图形类 (具有求面积和求周长的抽象方法)

派生类:圆形 矩形 (实现父类中的2个方法)

Main函数中定义图形数组 包含两个矩形1个圆形 并求出他们得总周长和总面积