

组号: 06



山东师范大学  
SHANDONG NORMAL UNIVERSITY

## 信息科学与工程学院课程实验报告

### 《面向对象程序设计》

姓名: 王汝芸

学号: 201711010202

班级: 计工本 1702

教师: 张庆科

时间: 2018/12/22

# 面向对象程序设计实验报告

姓名	王汝芸	班级	计工本 1702	学号	201711010202	组号	06
时间	2018/12/22	地点	山师长清	周次	17	页码	2
源码	<input checked="" type="checkbox"/> 无源码 <input type="checkbox"/> 文档源码 <input type="checkbox"/> 托管源码						
报告内容	<p><b>实验报告要求：</b>请围绕实验目的、实验内容、实验过程及步骤(可添加文字、矢量图)、实验结论与分析进行撰写。凡涉及源代码内容可给出完整源码并附上源码 Github 托管网址（请务必按照条目书写）。</p> <p><b>第六章知识总结</b></p> <p><b>第六章 多态性</b></p> <ul style="list-style-type: none"><li><b>1 多态概念与种类</b><ul style="list-style-type: none"><li>what: 多态性指的是同一种属性或行为对应着多种不同的实现。各个子类别对基类的实现方法彼此均不一样。</li><li>种类<ul style="list-style-type: none"><li>静态多态: 编译时匹配好要调用哪一个函数。<ul style="list-style-type: none"><li>函数重载</li><li>运算符重载</li></ul></li><li>动态多态: 运行时匹配好要调用哪一个函数。<ul style="list-style-type: none"><li>通过指针确定</li></ul></li></ul></li><li>重载函数<ul style="list-style-type: none"><li>原则: 同名不同参。在形参的个数, 顺序, 类型彼此存在不同。</li><li>种类<ul style="list-style-type: none"><li>非类的一般性函数重载</li><li>同一个类的成员函数重载</li><li>在基类和派生类中的同名成员函数重载</li></ul></li></ul></li></ul></li><li><b>2 静态多态的实现</b><ul style="list-style-type: none"><li>本质: 对已有的运算符赋予多重含义。同一运算符作用于不同类型的数据就会产生不同的形态。</li><li>作用: 经过重载后运算符可以直接用于类对象, 实现基本类型与类类型数据的统一快速运算。</li><li>形式<ul style="list-style-type: none"><li>成员函数: 将运算符重载定义为类的一个成员函数。&lt;函数类型&gt; operator &lt;运算符&gt; (第二操作数) 调用对象认为第一操作数。</li><li>友元函数: 将类外的一个运算符重载函数定义为类的一个友元函数。&lt;函数类型&gt; operator &lt;运算符&gt; (参数1, 参数2)。</li></ul></li><li>运算符重载<ul style="list-style-type: none"><li>基本原则: 江山易改本性难移。<ul style="list-style-type: none"><li>优先级与结合性都不会改变</li><li>功能与原有运算符功能相类似</li><li>不改变原有运算符的操作目标的个数, 同时至少要有有一个操作目标是用户自定义类型</li></ul></li></ul></li><li>常用运算符重载<ul style="list-style-type: none"><li>赋值运算符=</li><li>函数调用运算符()</li><li>输入输出运算符&lt;&lt;和&gt;&gt;</li><li>自增运算符++</li><li>自减运算符--</li><li>区分前后++</li><li>在后续运算符函数的形参量额外再添加一个int参数</li></ul></li><li>注意: 重载后的运算符的操作数都是类的对象, 非基本数据类型。</li><li>why: 无法通过基类指针或引用访问到派生类对象新增的成员函数。只在相应的函数声明时出现, 函数定义时不出现。</li><li>概念与声明<ul style="list-style-type: none"><li>概念: 在类的定义中, 前面有 virtual 关键字修饰的成员函数就是虚函数。</li><li>声明: virtual &lt;返回类型&gt; &lt;成员函数名&gt; (形式参数表);</li></ul></li><li>实现动态多态的前提条件<ul style="list-style-type: none"><li>public方式继承基类</li><li>同名函数原型完全相同</li></ul></li><li>本质作用: 实现动态多态。<ul style="list-style-type: none"><li>指针指向基类的对象, 调用基类虚函数</li><li>指针指向派生类的对象, 调用派生类虚函数</li><li>调用哪个虚函数取决于调用哪种类型的对象</li></ul></li><li>注意事项<ul style="list-style-type: none"><li>虚函数都是同名函数</li><li>虚函数是纯声明性函数</li><li>虚函数是非静态成员函数。因为静态成员函数不属于类的对象, 无法被继承。</li><li>派生类必须以公有方式继承基类</li><li>使用禁区<ul style="list-style-type: none"><li>静态成员函数不可设为虚函数。因为静态成员函数不属于某一个对象</li><li>内联函数不可设为虚函数。因为内联函数的执行代码是明确的, 没有多态性的特征</li><li>构造函数不可设为虚函数。对象还没有完全建立</li></ul></li></ul></li><li>虚析构函数<ul style="list-style-type: none"><li>why: 确保编译器只会调用基类析构函数, 派生类对象在内存里没有被完全释放。</li><li>声明: virtual ~类名()</li><li>一虚全虚: 如果基类的析构函数声明为虚函数, 则该类所有派生类的析构函数也自动成为虚函数, 而无需显式声明。</li></ul></li><li>虚函数与同名变量<ul style="list-style-type: none"><li>虚函数: 函数原型完全相。在基类的成员函数被声明为虚函数后, 其所有派生类中要有该同名虚函数的重新定义版本。</li><li>常虚函数: 函数原型不完全相同。调用时遵循同名变量。</li></ul></li><li>纯虚函数<ul style="list-style-type: none"><li>why: 基类往往是抽象的, 无法具象化成对象。</li><li>声明: 纯虚函数是一种只有函数声明没有函数体的虚函数。virtual 返回类型 成员函数名(形参) = 0;</li><li>注意事项<ul style="list-style-type: none"><li>无函数体</li><li>不能被调: 纯虚函数只有在派生类中被具体定义后, 才可被调用。</li><li>功能接口: 纯虚函数提供函数原型, 派生类给出具体实现代码。</li></ul></li></ul></li><li>抽象类<ul style="list-style-type: none"><li>定义: 包含纯虚函数的类称为抽象类。</li><li>理解: 有些类并没有包含足够的信息来描述一个具体的对象, 主要作为基类来使用。</li><li>注意事项<ul style="list-style-type: none"><li>抽象类可以有基类, 则其基类不能是普通类, 只能是抽象类。</li><li>抽象类必须至少含有一个纯虚函数, 也可以含有普通成员函数或静态函数。</li><li>抽象类可以定义指针或引用。</li><li>抽象类不能生成对象。</li></ul></li></ul></li></ul></li><li><b>4 纯虚函数与抽象类</b></li></ul>						

Ⓣ：可根据内容自行拓展页面，作业内容尾部尽量不要留有空白