第2讲（2）7-12章预习

《C++语言程序设计》在线直播

清华大学　郑　莉

目录

第7章 类的继承

继承机制的实际意义

- 从一些容易理解的实例出发，理解“继承”的意义
  - 概念的继承和代码的重用
  - 对新问题的适应

继承时的成员访问控制

- 继承过程中可以设定访问控制权限
- 无论以任何方式继承，基类的成员都存在于派生类的对象中。
- 基类的私有成员只能通过公有接口访问

继承时的构造、析构函数

- 构造函数和析构函数不被继承——基类的构造函数只能初始化从基类继承来的成员。派生类新增成员要自己设计构造函数来初始化
- 理解构造函数的执行次序和参数转递
  - 首先初始化基类子对象、对象成员，最后初始化派生类新增成员
  - 派生类要负责为基类的构造函数和对象成员的构造函数转递参数
  - 如果为一个类编写了构造函数，就最好再写一个默认的构造函数
- 析构函数的调用次序与构造函数相反

虚继承

- 多继承有可能导致冗余和二义性问题，可以通过虚继承解决
- 继承一个类时，很难预料将来是否会出现二义性，虚继承的基类及各派生类，直至最远派生类通常是由一个人或项目组一次设计完成的
- 能不用虚继承尽量不用，会给后继派生类增加复杂性

第8章 多态性

多态机制的背景和意义

- 要从我们熟悉的多态的思维方式入手理解程序对多态的模拟
- 从设计新的类型的角度理解运算符重载的意义——一个新的类型包括数据和运算方法
- 运算符重载实质上也就是函数重载，属静态多态性

运算符重载

- 这一章只讲运算符重载的一般概念和基本语法，对一些特殊运算符的重载，如[]，留到第9章结合实例学习会比较直观

动态多态性

- 绑定的概念
- 以基类的指针指向派生类的对象，并希望由此调用派生类的函数——静态绑定不能实现，使用动态绑定。

第9章 模板与群体类型

为什么介绍数据结构基础知

- 数据结构基础有利于对STL的理解
  - 在学习第一门语言时还没上数据结构课
- 借本章的一些综合例题，对全书内容作一个复习

模板

- 模板是C++语言的特色
- 这一章只介绍函数模板与类模板的基础知识，初步认识模板的程序设计思想
- 重点理解模板的意义和用途

数组类模板

- 安全性和可扩展性
- 使用模板达到通用性
- 借此例题综合复习
- 深层复制问题
- 特殊运算符重载

链表类模板

- 链表的基本概念
- 链表操作的基本方法
- 为学习STL打下基础

栈与队列的概念

- 着重理解栈与队列的概念和应用场合，以及栈与队列的基本操作方法
- 为理解STL中的类模板奠定基础

查找与排序算法

- 学习常用的查找与排序算法，能够解决一些简单的查找与排序问题，并有助于理解STL的算法
- 以函数模板来实现对各种不同类型数据群体的处理，这实际上就是泛型程序设计中通用算法的思想，有助于理解STL

第10章 泛型程序设计与STL

泛型程序设计思想

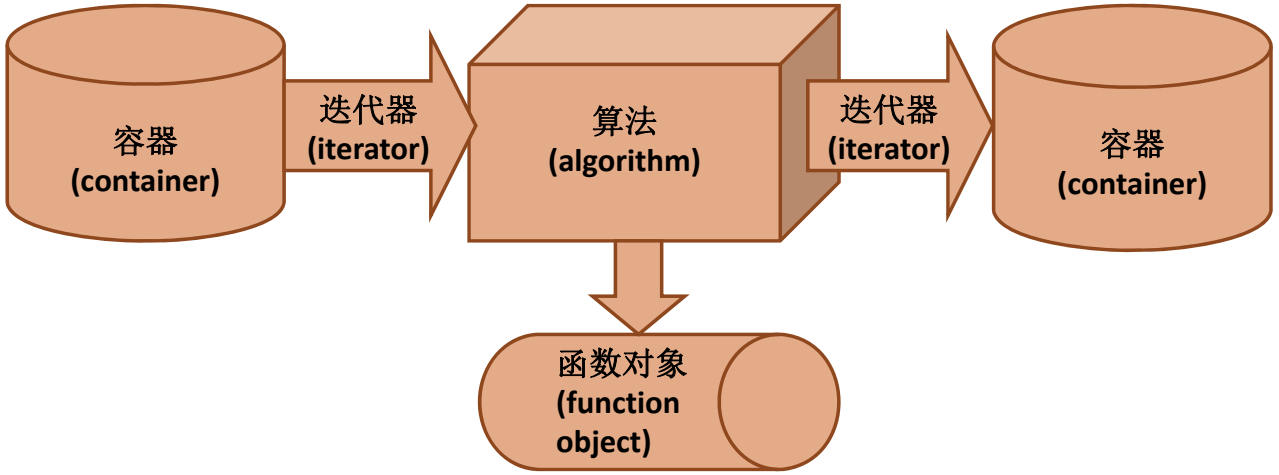
- 泛型程序设计思想不同于面向对象的思想，但是两者并不矛盾，配合使用可以使程序更为通用、灵活
- 这一章只介绍泛型程序设计的基本思想
  - 编写不依赖于具体数据类型的程序
  - 将算法从特定的数据结构中抽象出来，成为通用的
  - C++的模板为泛型程序设计奠定了关键的基础

STL简介

- 简单介绍STL的相关概念、术语，目的是使学生以后自学容易一些
- 主要概念是：容器、迭代器、适配器、通用算法

一张图概括STL

- Iterators（迭代器）是算法和容器的桥梁。
  - 将迭代器作为算法的参数、通过迭代器来访问容器而不是把容器直接作为算法的参数。
- 将函数对象作为算法的参数而不是将函数所执行的运算作为算法的一部分。
- 使用STL中提供的或自定义的迭代器和函数对象，配合STL的算法，可以组合出各种各样的功能。



第11章 输入/输出流与文件

主要内容

- I/O流的基本思想
- 基本的文件操作方法

学习重点

- 按专题来学习：文本文件、二进制文件、输入、输出
- 对每种问题学会一种方法就可以
- 首先从比较容易的题目体会到成功地读写文件的成就感，更多的I/O流类，在需要的时候可以自己看参考手册。

关于文本输出的格式问题

- 不用花太多时间
- 知道可以控制文本输出的格式就可以了，如果有时间，各种格式效果可以自己看参考手册去尝试

第12章 异常处理

主要内容

- 异常处理的基本思想
- C++的异常处理机制简介

重点

- 初学程序设计的学生，对程序的容错没有体会，没有在程序中检测错误、处理错误的意识
- 用过C等面向过程语言的学生，习惯随时用于if-else判断错误和就地处理

- 理解容错机制比学会语法重要
- 理解程序模块要有分工，理解异常处理的机制