广州大学学生实验报告

2018年4月7日

**开课学院及实验室：**计算机科学与工程实验室电子楼412A室

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **学院** | 计算机 | **专业班级** | 软件171 | **姓名** | 谢金宏 | **学号** | 1706300001 |
| **实验课程名称** | | 面向对象程序设计实验 | | **成绩** | |  | |
| **实验项目名称** | | 继承与多态性 | | **指导老师** | | 陶文正 | |

# 实验目的及要求

1. 掌握继承、基类和派生类的概念。
2. 掌握初始化基类成员的方法。
3. 掌握派生类对基类的继承。
4. 学习虚函数和纯虚函数的定义与使用方式。
5. 理解抽象类的概念，学习如何用指针指向其他的派生类，实现多态性。
6. 掌握抽象类的定义与使用方式，并注意指针的用法。
7. 学习如何使用虚函数、纯虚函数、抽象类和实现类的多态性。

# 实验设备及平台

1. 硬件环境：计算机
2. 软件环境：G++ 7.2.0, Visual Studio Code

# 实验内容及步骤

1. **定义派生类的实例**

#include <string>

#include <iostream>

using namespace std;

class Student {

protected:

string serial, name;

int sex, year, month, day;

public:

Student() = default;

Student(string serial, string name, int sex, int year, int month, int day) {

this->serial = serial;

this->name = name;

this->sex = sex;

this->year = year;

this->month = month;

this->day = day;

}

void SetSerial(string serial) {

this->serial = serial;

}

void SetName(string name) {

this->name = name;

}

void SetSex(int sex) {

this->sex = sex;

}

void SetYear(int year) {

this->year = year;

}

void SetMonth(int month) {

this->month = month;

}

void SetDay(int day) {

this->day = day;

}

string GetSerial() {

return serial;

}

string GetName() {

return name;

}

int GetSex() {

return sex;

}

int GetYear() {

return year;

}

int GetMonth() {

return month;

}

int GetDay() {

return day;

}

};

class ExtendedStudent: public Student {

protected:

int C, English;

public:

ExtendedStudent() = default;

ExtendedStudent(string serial, string name, int sex, int year, int month, int day, int C, int English)

: Student(serial, name, sex, year, month, day) {

this->C = C, this->English = English;

}

void SetC(int C) {

this->C = C;

}

void SetEnglish(int English) {

this->English = English;

}

int GetC() {

return C;

}

int GetEnglish() {

return English;

}

void List() {

cout << "学号：" << serial << endl;

cout << "姓名：" << name << endl;

cout << "性别：" << (sex == 0 ? "女" : "男") << endl;

cout << "出生日期：" << year << "年" << day << "月" << day << "日" << endl;

cout << "C++：" << C << endl;

cout << "英语：" << English << endl;

}

};

template <typename T>

void input(T &t, const string hint)

{

cout << hint; cin >> t;

}

int main() {

int n;

cout << "请输入学生人数：" << endl;

cin >> n;

ExtendedStudent \*s = new ExtendedStudent[n];

for (int i=0; i<n; ++i) {

cout << "正在输入第" << i+1 << "位学生" << endl;

string serial, name;

int sex, year, month, day, C, English;

input(serial, string("学号："));

input(name, string("姓名："));

input(sex, string("性别（非零数字表示男性）："));

cout << "出生年月日：" << endl;

cin >> year >> month >> day;

input(C, string("C语言成绩："));

input(English, string("英语成绩："));

s[i] = ExtendedStudent(serial, name, sex, year, month, day, C, English);

}

cout << endl;

cout << "下面列出已输入学生的信息：" << endl;

for (int i=0; i<n; ++i) {

if (i) cout << endl;

cout << "第" << i+1 << "位学生" << endl;

s[i].List();

}

delete[] s;

}

以上程序提醒输入学生的数量和对应数量学生的信息；输入完毕后将列出输入学生的信息。程序使用了面向对象编程中的继承特性。

1. **多重继承的实例**

#include <iostream>

using namespace std;

class Base1 {

public:

    void show\_i() { cout << i << endl; }

protected:

    int i;

};

class Base2 {

public:

    void show\_j() { cout << j << endl; }

protected:

    int j;

};

class Derived : public Base1, public Base2 {

public:

    void set(int x, int y) { i = x, j = y; }

};

int main() {

    Derived obj;

    obj.set(5, 7);

    obj.show\_i();

    obj.show\_j();

}

以上程序是多重继承的一个例子。

1. **抽象并编程现实生活中图书馆借阅的相关内容**

设有“借阅者”类，主要功能是记录借书证号以及借阅书籍，并记录借还书的日期等；“学生”类，记录学生的姓名和学号；设计“学生借阅者”类，它是“借阅者”类和“学生”类的派生类。在此基础上设计查找函数，根据输入的学生借阅者姓名，显示出该借阅者的相关信息。

# 思考问题及课后练习

# 总结