

# 中山大学移动信息工程学院本科生实验报告

(2015-2016 学年春季学期)

课程名称: Data structures and algorithms

任课教师: 张子臻、黄淦

年级	15	专业(方向)	软件工程(移动信息工程)
学号	15352408	姓名	张稼伟
电话	13531810182	Email	709075442@qq.com
开始日期	2016. 4. 25	完成日期	2016. 4. 25

## 1. 实验题目

**1000:**

Bank 类: 存储银行账户信息

ICBC 类: Bank 的派生类, 存储 ICBC 银行账户信息。

ABC 类: Bank 的派生类, 存储 ABC 银行账户信息。

设计一个 AsscountManage 类, 来处理各种银行的账单, 包括存钱, 取钱, 不同银行账户之前的转账。

## 2. 实验目的

- A. 加深对面向对象编程的理解。
- B. 练习对继承的运用并体会继承的特点及好处。
- C. 学会使用纯虚函数并知道抽象类的特点。
- D. 学习使用二级指针。

## 3. 程序设计

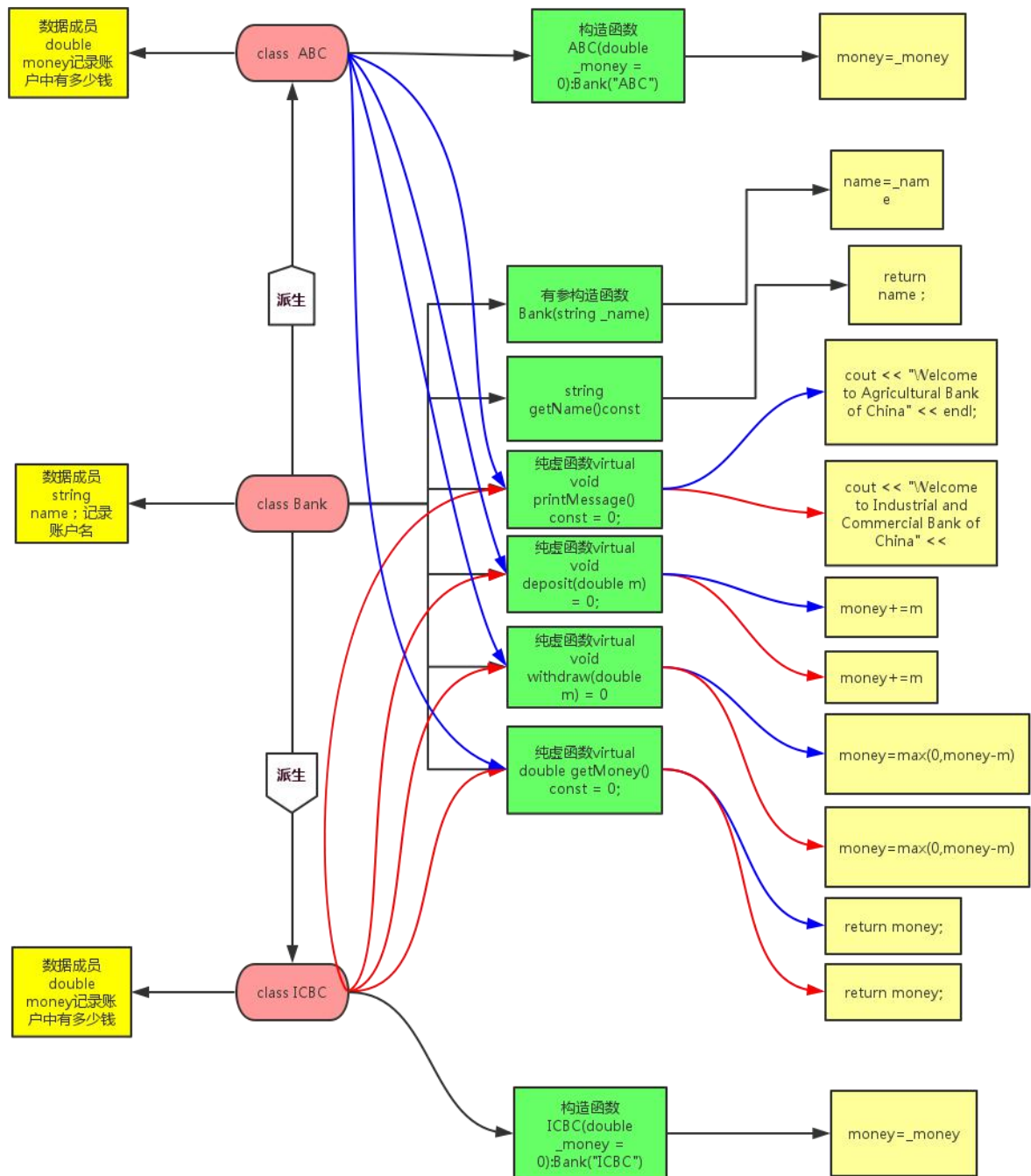
**1000:**

**注意事项:** A. 由于 bankList 的实际作用是存储指针的数组, 所以我们在析构函数中应该先 delete 掉每个 bankList[i] 再 delete bankList;

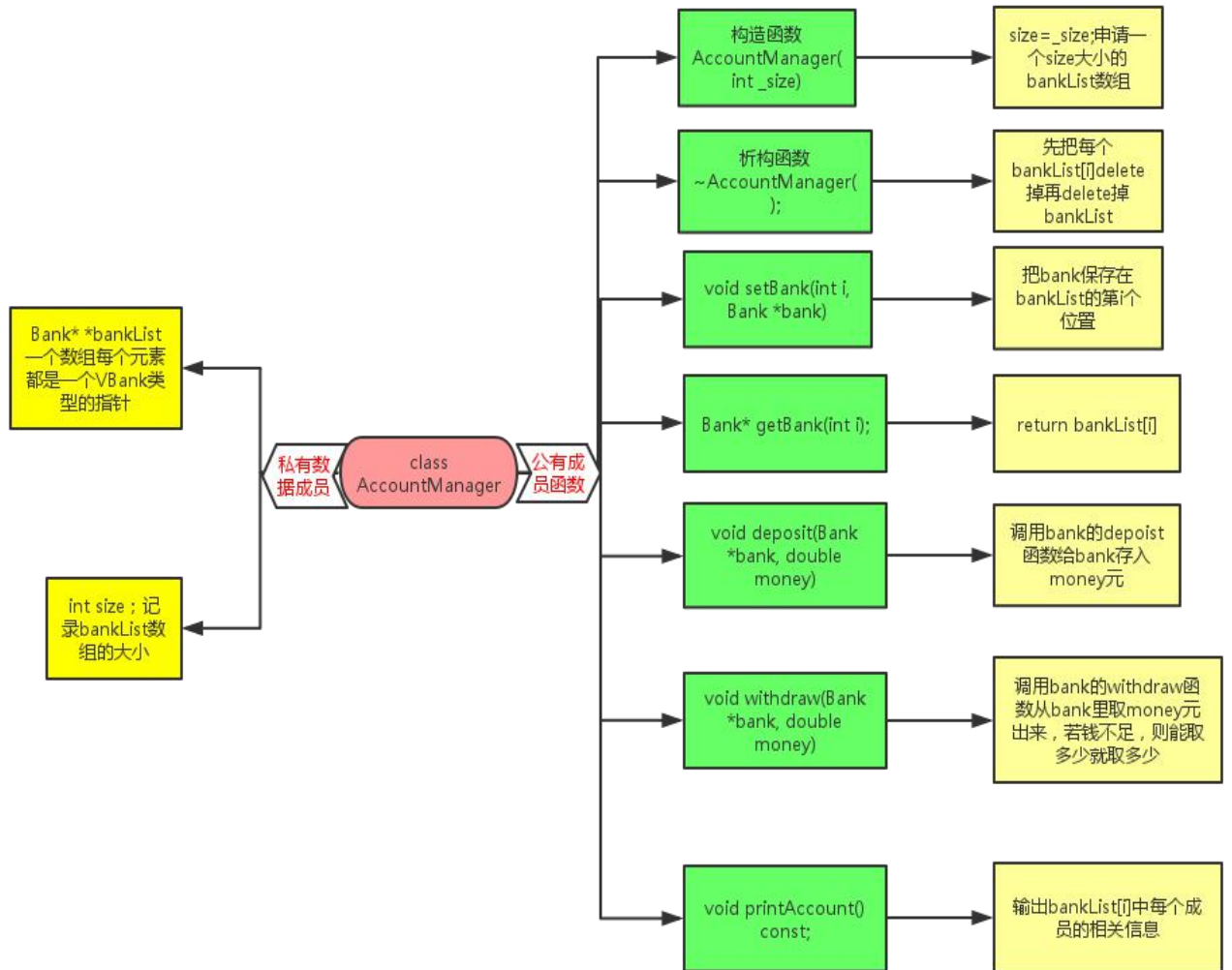
B. 注意取钱或转账时应先判断钱够不够，若不够，能用多少就用多少钱。

流程图：

Bank 类，ICBC 类，ABC 类：



## AccountManager 类:



本题完整可测试代码详见附件 1000.cpp

---

## 4. 程序运行与测试

1000:

测试一:

```
AccountManager accountManager(2);

Bank *bank1 = new ICBC(100);
accountManager.setBank(0, bank1);
Bank *bank2 = new ABC;
accountManager.setBank(1, bank2);

accountManager.printAccount();

accountManager.deposit(accountManager.getBank(0), 100);
accountManager.deposit(accountManager.getBank(1), 100);
accountManager.printAccount();

accountManager.withdraw(accountManager.getBank(0), 50);
accountManager.withdraw(accountManager.getBank(1), 50);
accountManager.printAccount();

accountManager.transfer(accountManager.getBank(0), accountManager.getBank(1), 500);
accountManager.printAccount();
```



```
Welcome to Industrial and Commercial Bank of China
ICBC 100
Welcome to Agricultural Bank of China
ABC 0
Welcome to Industrial and Commercial Bank of China
ICBC 200
Welcome to Agricultural Bank of China
ABC 100
Welcome to Industrial and Commercial Bank of China
ICBC 150
Welcome to Agricultural Bank of China
ABC 50
Welcome to Industrial and Commercial Bank of China
ICBC 0
Welcome to Agricultural Bank of China
ABC 200
```

---

## 测试二:

```
AccountManager accountManager(2);

Bank *bank1 = new ICBC (89);
accountManager.setBank(0, bank1);
Bank *bank2 = new ABC(12);
accountManager.setBank(1, bank2);

accountManager.printAccount();

accountManager.deposit(accountManager.getBank(0), 21);
accountManager.deposit(accountManager.getBank(1), 88);
accountManager.printAccount();

accountManager.withdraw(accountManager.getBank(0), 79);
accountManager.withdraw(accountManager.getBank(1), 100);
accountManager.printAccount();

accountManager.transfer(accountManager.getBank(0), accountManager.getBank(1), 2);
accountManager.printAccount();
```



```
Welcome to Industrial and Commercial Bank of China
ICBC 89
Welcome to Agricultural Bank of China
ABC 12
Welcome to Industrial and Commercial Bank of China
ICBC 110
Welcome to Agricultural Bank of China
ABC 100
Welcome to Industrial and Commercial Bank of China
ICBC 31
Welcome to Agricultural Bank of China
ABC 0
Welcome to Industrial and Commercial Bank of China
ICBC 29
Welcome to Agricultural Bank of China
ABC 2
```



### 测试三:

```
AccountManager accountManager(2);

Bank *bank1 = new ICBC(10);
accountManager.setBank(0, bank1);
Bank *bank2 = new ABC(150);
accountManager.setBank(1, bank2);

accountManager.printAccount();

accountManager.deposit(accountManager.getBank(0), 200);
accountManager.deposit(accountManager.getBank(1), 200);
accountManager.printAccount();

accountManager.withdraw(accountManager.getBank(0), 188);
accountManager.withdraw(accountManager.getBank(1), 600);
accountManager.printAccount();

accountManager.transfer(accountManager.getBank(0), accountManager.getBank(1), 100);
accountManager.printAccount();
```

```
Welcome to Industrial and Commercial Bank of China
ICBC 10
Welcome to Agricultural Bank of China
ABC 150
Welcome to Industrial and Commercial Bank of China
ICBC 210
Welcome to Agricultural Bank of China
ABC 350
Welcome to Industrial and Commercial Bank of China
ICBC 22
Welcome to Agricultural Bank of China
ABC 0
Welcome to Industrial and Commercial Bank of China
ICBC 0
Welcome to Agricultural Bank of China
ABC 22
```

---

## 5. 实验总结与心得

1. 含有纯虚函数的类叫抽象类，抽象类是一种特殊的类，它是为了抽象和设计的目的而建立的，它往往用来表征对问题领域进行分析、设计中得出的抽象概念，它不能实例化。
2. 抽象类只能用作基类，其纯虚函数的实现由派生类给出。注意派生类要给出基类中所有纯虚函数的实现，否则这个派生类还是一个抽象类。
3. 可以定义指向抽象类的指针和引用，此指针可以指向它的派生类进而实现多态性。
4. 本题中 AccountManager 类的数据成员 bankList 是一个 Bank 类型的二级指针。二级指针实际上就是指向指针的指针。我们知道任何值都有地址，一级指针的值虽然是一个地址，但是这个地址也需要一个空间存放，是空间就有地址，这就是存放地址这一值得空间所具有的地址，二级指针的值就是这个地址。
5. 二级指针分为指向指针变量的指针和指向数组的指针。所谓指向指针变量的指着就如第 4 点所述。我们知道一级指针指向数组时，就可以直接通过指针对数组进行操作，这里二级指针就指向这个一级指针。本题中的 bankList 实际上是一个指针数组，数组中每一个元素是一个 Bank 类型的一级指针，而这个数组又通过一个指针来操作。

## 附录、提交文件清单

实验报告一份：实验报告.pdf

代码一份：1000.cpp