**实验十三 类的定义与使用**

1. **实验目的和要求**

编写一个程序，定义一个银行系统的账户类SavingAccount。每个账户包含的信息有账号、存入日期、存款金额和月利率。要求账号自动生成，第一个生成的对象账号为1，第二个生成的对象账号为2，以此类推。所需的操作包括修改月利率、每月计算新的存款额（原存款额+本月利息）和显示账户金额。

**二、实验内容**

1. 实验准备
   1. 理论知识介绍
      1. 数据的输入：

要从用户依次读入int类型的变量a,n，可以使用如下语句:

cin >> a>>n;

* + 1. 数据的输出

要将变量的内容显示在显示器上，可以使用cout和流插入运算符。Cout还可以输出表达式的执行结果，比如:

cout << a << ‘+’ << b << ‘=’ << a+b << endl;

* + 1. 算术表达式

C++中，算术运算里有+（加法），-（减法），/（除法），%（取模），但是没有取平方算数运算。

* + 1. 数据类型 （整型）：

C++中一个整型变量可以储存一个整数，可以直接通过cin和cout输入和输出。

* + 1. 函数

函数定义的形式如下：

类型名 函数名（形式参数列表）

{

变量定义部分

语句部分

}

* + 1. 类的定义

定义一个类就是定义一组属性和一组堆属性进行操作的函数。属性称为类的数据成员，而函数称为类的成员函数。类定义的一般形式如下：

class 类名{

private:

私有数据成员和成员函数;

public:

公有数据成员和成员函数;

};

* + 1. 对象的使用

在程序中直接定义对象的方法与定义普通函数的方法一样，他的格式如下：

存储类别 类名 对象列表;

对象可以用如下的语句操作：

结构体变量名.成员名

* + 1. 对象的构造

有了构造函数之后，对象的定义的一般形式如下：

类名 对象名(实际参数表)

* + 1. 对象的析构

一般析构函数的定义是：

~类名() {

函数体语句

}

* + 1. 静态成员

类的静态数据成员拥有一块单独的存储区，而不管创建了多少个该类的对象。所有这些对象的静态数据成员都共享这一块空间。这就为这些对象提供了一种相互通信的机制。

要讲一些数据成员说明为静态的，只需要在此数据成员之间加一个保留关键词static。

1. 实验项目
   1. 分析
      1. 由于对于每一个账户，月利率都是一样的，所以我们可以将月利率设置为一个静态的成员变量。
      2. 由于要求每一个账户的账号要自动生成，所以一定有一个变量来储存账户的数量，存储这个数量的变量也可以用静态变量来实现。
   2. 方案
      1. 需要提供两个文件，一个为saving\_account.h,另一个为saving\_account.cpp,分别包含SavingAccount类的定义和实现
      2. saving\_account.cpp里应该还包含一个main函数，用来测试SavingAccount类
      3. SavingAccount类应该包括月利率，账户个数，账号，存款日期，存款金额这些私有成员，还应该包括构造函数，显示金额函数，刷新存款额函数
      4. 由于SavingAccount类中没有动态的内存分配，所以我们不需要自己定义析构函数
   3. 测试数据

SavingAccount account1("20170516", 1000);

SavingAccount account2("20170517", 2000);

SavingAccount account3("20170518", 3000);

account1.Display();

account1.RefreshDeposit();

account1.Display();

account2.Display();

account3.Display();

account1.SetRate(0.02);

account1.Display();

account2.RefreshDeposit();

account2.Display();

return 0;

* 1. 运行结果

The deposit is 1000.

The deposit is 1010.

The deposit is 2000.

The deposit is 3000.

The deposit is 1010.

The deposit is 2040.

**三、实验小结**

1. 完成情况

已经完成了要求的功能，但是程序的整洁性不够好。

1. 重点

类的定义，类的构造函数，类的静态变量。

1. 难点

利用类的静态变量进行对象之间的共享信息。

1. 解决方法

在类的函数中可以使用SavingAccount的静态变量count\_来对其非晶态变量account\_赋值。

1. 有待改进之处

我们可以在SavingAccount这个类中加一个时间的静态变量，再设计一个自动的更新存款金额的函数，自动检查时间从而自动每一个月增加存款金额。

1. 收获、体会

类的静态成员不仅可以让整个程序所占的内存变小，还能让多个对象之间共享数据，给程序提供了极大的灵活性。