```
    cout << a.getX() << "    " << a.getY() << endl;
    p = q;
    cout << p->getX() << "    " << p->getY() << endl;

cout << c.getX() << "    " << d.getX() << "    " << d.getY() << endl;

    b = a;
    cout << b.getX() << "    " << b.getY() << endl;
}
```

Homework:

12.3　12.7

Lab5 Object-Oriented Programming:Polymorphism

Objectives

1. What polymorphism is, how it makes programming more convenient, and how it makes systems more extensible and maintainable.
2. To declare and use virtual functions to effect polymorphism.
3. The distinction between abstract and concrete classes.
4. To declare pure virtual functions to create abstract classes.
5. How C++ implements virtual functions and dynamic binding "under the hood."
6. How to use virtual destructors to ensure that all appropriate destructors run on an object.

Experiment

Ex 1：（习题 13.12， Employee 类继承层次）

1. Description of the Problem

英文：(Payroll System Modification) Modify the payroll system of Figs. 13.13~13.23 to include private data member birthDate in class Employee. Use class Date from Figs. 11.12~11.13 to represent an employee's birthday. Assume that payroll is processed once per month. Create a vector of Employee references to store the various employee objects. In a loop, calculate the payroll for each Employee (polymorphically), and add a $100.00 bonus to the person's payroll amount if the current month is the month in which the Employee's birthday occurs.

中文：修改图 13.13~13.23 的工资系统，增加 private 数据成员 birthDate(Date 对象), 要求使用图 11.12~11.13 的 Date 作为生日类型。假设工资系统每月处理一次，创建一个 vector 存储 Employee 指针来存储不同的员工对象，用一个循环计算每个员工的工资时(多态)，遇到当月过生日的员工多发 100 美元奖金。

2. Problem-Solving Tips

1）取当前时间函数提示：

方法一：

```
#include <windows.h>
int    main()
{ SYSTEMTIME systm;
   GetLocalTime(&systm);
   cout<<systm.wYear<<"-"<<systm.wMonth<<"-"<<systm.wDay<<" "<<
        systm.wHour<<":"<<systm.wMinute<<":"<<systm.wSecond;
 return 0;
}
```

方法二：

```
#include <iostream>
#include <ctime>
using namespace std;
int    main()
{
 time_t nowtime;
 struct tm* ptm;
 time(&nowtime);
 ptm = localtime(&nowtime);
 cout<<ptm->tm_year + 1900<<"-"<<ptm->tm_mon + 1<<"-"<<ptm->tm_mday<<"    "
 <<ptm->tm_hour<<":"<<ptm->tm_min<<":"<<ptm->tm_sec;
 return 0;
}
```

2）测试函数示例

参考教材的测试函数 13.23 和 13.25。

3. 结果示例

```
Employees processed polymorphically via dynamic binding:

salaried employee: John Smith
birthday: June 15, 1944
social security number: 111-11-1111
weekly salary: 800.00
earned $800.00

hourly employee: Karen Price
birthday: April 29, 1960
social security number: 222-22-2222
hourly wage: 16.75; hours worked: 40.00
HAPPY BIRTHDAY!
earned $770.00

commission employee: Sue Jones
birthday: September 8, 1954
social security number: 333-33-3333
gross sales: 10000.00; commission rate: 0.06
earned $600.00

base-salaried commission employee: Bob Lewis
birthday: March 2, 1965
social security number: 444-44-4444
gross sales: 5000.00; commission rate: 0.04; base salary: 300.00
earned $500.00

deleting object of class SalariedEmployee
deleting object of class HourlyEmployee
deleting object of class CommissionEmployee
```

Ex 2：（习题 13.13，多态，Account 类继承层次）

1. Description of the Problem

(Shape Hierarchy) Implement the Shape hierarchy designed in Exercise 12.7 (which is based on the hierarchy in Fig. 12.3). Each TwoDimensionalShape should contain function getArea to calculate the area of the two-dimensional shape. Each ThreeDimensionalShape should have member functions getArea and getVolume to calculate the surface area and volume of the three-dimensional shape, respectively. Create a program that uses a vector of Shape pointers to objects of each concrete class in the hierarchy. The program should print the object to which each vector element points. Also, in the loop that processes all the shapes in the vector, determine whether each shape is a TwoDimensionalShape or a ThreeDimensionalShape. If a shape is a TwoDimensionalShape, display its area. If a shape is a ThreeDimensionalShape, display its area and volume.

2. 实验要求

（1）画出类图，写够 5 个类定义，并且继承关系正确

（2）每个类的成员函数定义，且必须至少包含以下函数：

构造函数，析构函数，virtual getArea 函数

（4）main 函数：

vector 数组声明

使用类对象对 vector 数组的赋值

提供一个测试函数 test，用来测试 getArea，其参数是基类指针或引用变量

Ex 3：（习题 13.16，多态，Account 类继承层次）

1. Description of the Problem

对实验 8 中的习题 12.10 进行修改。

创建一个与银行账户相关的类继承层次。银行的所有账户都可以存款和取款。存款能够产生一定的利息，查询和取款交易要缴纳一定的手续费。

要求：基类：Account(参考实验提示)；

派生类：SavingAccount 和 CheckingAccount；

SavingAccount：继承 Account 的成员函数；构造函数接收两个参数：存款初始值（initialBalance）和利率（rate）；增加一个数据成员：利率(interestRate)，增加 public 类型的成员函数用于计算利率(calculateInterest())。

CheckingAccount：构造函数应接收到两个参数，一个是存款初始值（initialBalance），一个是手续费（fee）；增加一个数据成员：手续费（transactionFee）；重新定义成员函数 credit()和 debit()，以便能够从存款余额中减去手续费，要求成员函数通过调用基类的成员函数来更新存款数目，debit()函数应该在确定取款之后才能扣除手续费。

要求 1：创建一个 vector 存储一组 SavingAccount 和 CheckingAccount 对象（多态），处理每一个账户时，判断该账户的类型，如果是 SavingAccount，使用其成员函数 calculateInterest()计算利率并加入账户，处理完一个账户，调用基类的成员函数 getBalance() 打印其新的存款。

2. Problem-Solving Tips

1）类定义：

要求独立完成。

2）测试函数：

要求独立完成。

3. 结果示例

```
Account 1 balance: $25.00
Enter an amount to withdraw from Account 1: 10
Enter an amount to deposit into Account 1: 30
Adding $1.35 interest to Account 1 (a SavingsAccount)
Updated Account 1 balance: $46.35

Account 2 balance: $80.00
Enter an amount to withdraw from Account 2: 0
$1.00 transaction fee charged.
Enter an amount to deposit into Account 2: 10
$1.00 transaction fee charged.
Updated Account 2 balance: $88.00

Account 3 balance: $200.00
Enter an amount to withdraw from Account 3: 10
Enter an amount to deposit into Account 3: 0
Adding $2.85 interest to Account 3 (a SavingsAccount)
Updated Account 3 balance: $192.85

Account 4 balance: $400.00
Enter an amount to withdraw from Account 4: 0
$0.50 transaction fee charged.
Enter an amount to deposit into Account 4: 0
$0.50 transaction fee charged.
Updated Account 4 balance: $399.00
```

Lab6 Templates

Objectives：