实验九：多态&文件&模板&异常

实验目的

1．掌握虚函数、多态、静态绑定和动态绑定
2．掌握函数模板和函数模板的特化、类模板和类模板的特化
3．掌握顺序文件的读写
4．理解异常的处理机制，掌握如何声明新的异常

实验作业

作业一（习题 12.12， Employee 类继承层次）

1. 问题描述

Modify the payroll system of Figs. 12.9–12.17 to include private data member **birthDate** in class **Employee**. Use class **Date** from Figs. 10.6–10.7 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.

2. 实验提示

取当前时间函数：
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;
}
```

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
```

作业二（顺序文件的读写）

1. 问题描述

用 for 结构为 ASCII 字符集中 ASCII 码值从 33~126 的字符打印出一张 ASCII 码表到一个**顺序文件**“ascii.txt”中。要求输出十进制值、八进制值、十六进制值和 ASCII 码值，并在程序中使用流操纵算子 dec、oct 和 hex。写入文件后，再次从文件中读取每一行并打印到屏幕。

2. 实验提示

表头打印：

```
cout << setw( 7 ) << "Decimal" << setw( 9 ) << "Octal " << setw( 15 )
    << "Hexadecimal " << setw( 13 ) << "Character"
    << showbase << '\n';
```

3. 结果示例

结果的部分拷屏如下图，实际结果应显示 33~126 之间的字符。

```
Decimal    Octal    Hexadecimal    Character
     33      041            0x21            !
     34      042            0x22            "
     35      043            0x23            #
     36      044            0x24            $
     37      045            0x25            %
     38      046            0x26            &
     39      047            0x27            '
     40      050            0x28            (
     41      051            0x29            )
     42      052            0x2a            *
     43      053            0x2b            +
     44      054            0x2c            ,
     45      055            0x2d            -
     46      056            0x2e            .
     47      057            0x2f            /
     48      060            0x30            0
     49      061            0x31            1
     50      062            0x32            2
     51      063            0x33            3
     52      064            0x34            4
     53      065            0x35            5
     54      066            0x36            6
     55      067            0x37            7
     56      070            0x38            8
```

作业三（Array 类模板）

1. 问题描述

修改图 10.10~11 的 Array 类，写一个类模板 Array，增加一个**非类型参数**：int 类型的普通参数，表示数组的大小。此类模板支持任意基本数据类型的元素，测试将类模板特化为 int 和 float 元素，并测试成员函数。

2. 结果示例

```
Enter 5 integer values:
1 2 3 4 5

The values in intArray are:
1 2 3 4 5

Enter 5 floating point values:
3.4 1.0 2.6 3.0 34

The values in the doubleArray are:
3.4 1 2.6 3 34
```

作业四（异常处理的逻辑流程）

1. 问题描述

Suppose a program throws an exception and the appropriate exception handler begins executing. Now suppose that the exception handler itself `throws` the same type of exception. Does this create infinite recursion? Write a program to check your observation.

2. 实验提示

a) 定义一个 runtime_error 派生类

class TestException : public runtime_error{}

b) main 函数

参考教材的 main 函数 16.2，在 try 语句块中抛出异常，并且在异常处理部分重新抛出该异常。

3. 结果示例

```
This is a test

abnormal program termination
```

作业五（构造函数、析构函数和异常处理）

1. 问题描述

Write a program illustrating that member object destructors are called for only those member objects that were constructed before an exception occurred.

2. 实验提示

a) 定义类 Item，并包含整型成员变量 value，并在 Item 的构造函数中定义条件判断语句以抛出异常，例如：

   if (value == 3) throw runtime_error("An exception was thrown");

b) main 函数

   main 函数中构建若干 Item 对象，并在合适位置打印测试语句。

3. 结果示例

```
Constructing an object of class ItemGroup
Item 1 constructor called
Item 2 constructor called
Item 3 constructor called
Item 2 destructor called
Item 1 destructor called
An exception was thrown
```