浙江理工大学信息电子学院

实验指导书

实验名称：类的继承机制的实现　　**学时安排**：3

实验类别：设计性实验　　**实验要求**：1人1组

学号：　　2012329620003　　姓名　：李畅

一、实验目的

1.掌握单继承和多重继承的概念。

2.理解不同的继承类型：public、protected和private，掌握何时使用何种继承类型。

3.掌握类层次中构造函数的定义方式和建立对象时构造和析构次序

二、实验原理介绍

通过继承机制实现对类功能的扩展,合理设计派生类的构造函数、成员函数。

三、实验设备介绍

软件需求：Visual C++ 6.0

硬件需求：对于硬件方面的要求，建议配置是Pentium III 450以上的CPU处理器，64MB以上的内存，200MB的自由硬盘空间、CD-ROM驱动器、能支持24位真彩色的显示卡、彩色显示器、打印机。

四、实验内容

实现对第一次实验结果Elevator类的功能扩展。在Elevator类已有功能的基础上派生ElevatorFactory类。ElevatorFactory类可以实现当多人在不同楼层等待乘坐上行或下行的同一部电梯时,能够合理的根据乘坐人的需求对电梯经停的楼层进行排序。

要求：

1.为了实现上的方便性，我们假设同一组要求乘坐电梯的乘客或者都是上行,或者都是下行。

2. 在主函数中对该类的功能进行测试，测试方法是首先选择在某一时间段一组要乘坐电梯的乘客是上行还是下行，然后输入组中乘客的人数及每一个乘客所在楼层和目的楼层，由 ElevatorFactory 类实例化后的电梯对象在运作的过程中，如果电梯是上行，则能根据乘客所在的楼层和目的楼层从下向上依次停靠；如果电梯是下行，则能根据乘客所在的楼层和目的楼层从上向下依次停靠。

3. 在测试的过程中，还需要注意测试当多个用户在同一楼层或多个用户的目的楼层为同一楼层时情况的处理。

程序要求的结果请运行 elevator.exe

提示：

为了方便描述乘客，我们可以定义一个 Person 类，主要描述每一个乘客所在楼层和目的楼层。ElevatorFactory 类从 Elevator 类继承而来，它从某一个时间段要乘坐电梯的每个乘客的信息当中提取其所在楼层和目的楼层信息，然后对它们进行排序，再由继承自基类 Elevator 的成员 setFloorNumber 对要停靠的楼层序列依次输出。

思考（可选）

如果加入乘客的体重信息，如何实现在停靠楼层对超载信息的提示。

五 程序清单

```
# include <iostream>
using namespace std;
class person{
public:
      int level1[20];
      int level2[20];
       int i;
    void input1(int num){
        for( i=0;i<num;i++){
            cout<<"请输入所在的楼层："<<endl;
            cin>>level1[i];
        }
```

```
    }
    void input2(int num){
        for( i=0;i<num;i++){
            cout<<"请输入所去的楼层："<<endl;
            cin>>level2[i];
        }
    }
};
class elevator{
public:
int j;
    int nowlevel,nextlevel,max,num;
    void up(int a){
             nextlevel=a;
            for(j=nowlevel;j<nextlevel;j++){
                cout<<endl<<"--"<<j<<"--"<<endl<<endl;
            }
            cout<<"第"<<nextlevel<<"层到了"<<endl;
            nowlevel=nextlevel;
    }
    void down(int a){
         nextlevel=a;
            for(j=nowlevel;j>nextlevel;j--){
                cout<<endl<<"--"<<j<<"--"<<endl<<endl;
            }
            cout<<"第"<<nextlevel<<"层到了"<<endl;
            nowlevel=nextlevel;
        }
}ele;
```

```
class ElevatorFactory:public elevator,public person{
public:
int k,temp;
    void up2(int num)
    {
          for(k=0;k<num;k++)
          {
            for(i=k+1;i<num;i++)
            {
                if(level1[i]<level1[k]){
                   temp=level1[k];
                   level1[k]=level1[i];
                   level1[i]=temp;
               }
               }
           }
         for(k=0;k<num;k++)
          {
            for(i=k+1;i<num;i++)
            {
                if(level2[i]<level2[k]){
                   temp=level2[k];
                    level2[k]=level2[i];
                   level2[i]=temp;
               }
               }
           }
                       nowlevel=level1[0];
           for(i=0;i<num;i++)
```

```
        {up(level2[i]);}
    }
    void down2(int num)
{
      for(k=0;k<num;k++)
      {
        for(i=k+1;i<num;i++)
        {
            if(level1[i]>level1[k]){
               temp=level1[k];
               level1[k]=level1[i];
               level1[i]=temp;
           }
           }
         }
      for(k=0;k<num;k++)
      {
        for(i=k+1;i<num;i++)
        {
            if(level2[i]>level2[k]){
               temp=level2[k];
               level2[k]=level2[i];
               level2[i]=temp;
           }
           }
         }
         nowlevel=level1[0];
         for(i=0;i<num;i++)
         {down(level2[i]);}
```

```
        }

};
void main(){
    ElevatorFactory elevator;
    int i,num,k;
    char choice;
        cout<<"此电梯一共十层"<<endl;
        cout<<"现在是一楼"<<endl;
        do{
        cout<<"-----请选择操作-----:"<<endl;
        cout<<endl<<endl;
        cout<<"  1、上升请按 1;2、下降请按 2"<<endl;
        cout<<endl<<endl;
        cout<<"---------------"<<endl;
        cin>>choice;
        cout<<"请输入乘坐电梯人数"<<endl;
        cin>>num;
        switch(choice){
            case'1':
                    {elevator.input1(num);
                            elevator.input2(num);
                    elevator.up2(num);
                    cout<<"1.继续,2.退出"<<endl;
                    cin>>k;
                break;}
            case'2': {elevator.input1(num);
                            elevator.input2(num);
                    elevator.down2(num);
```

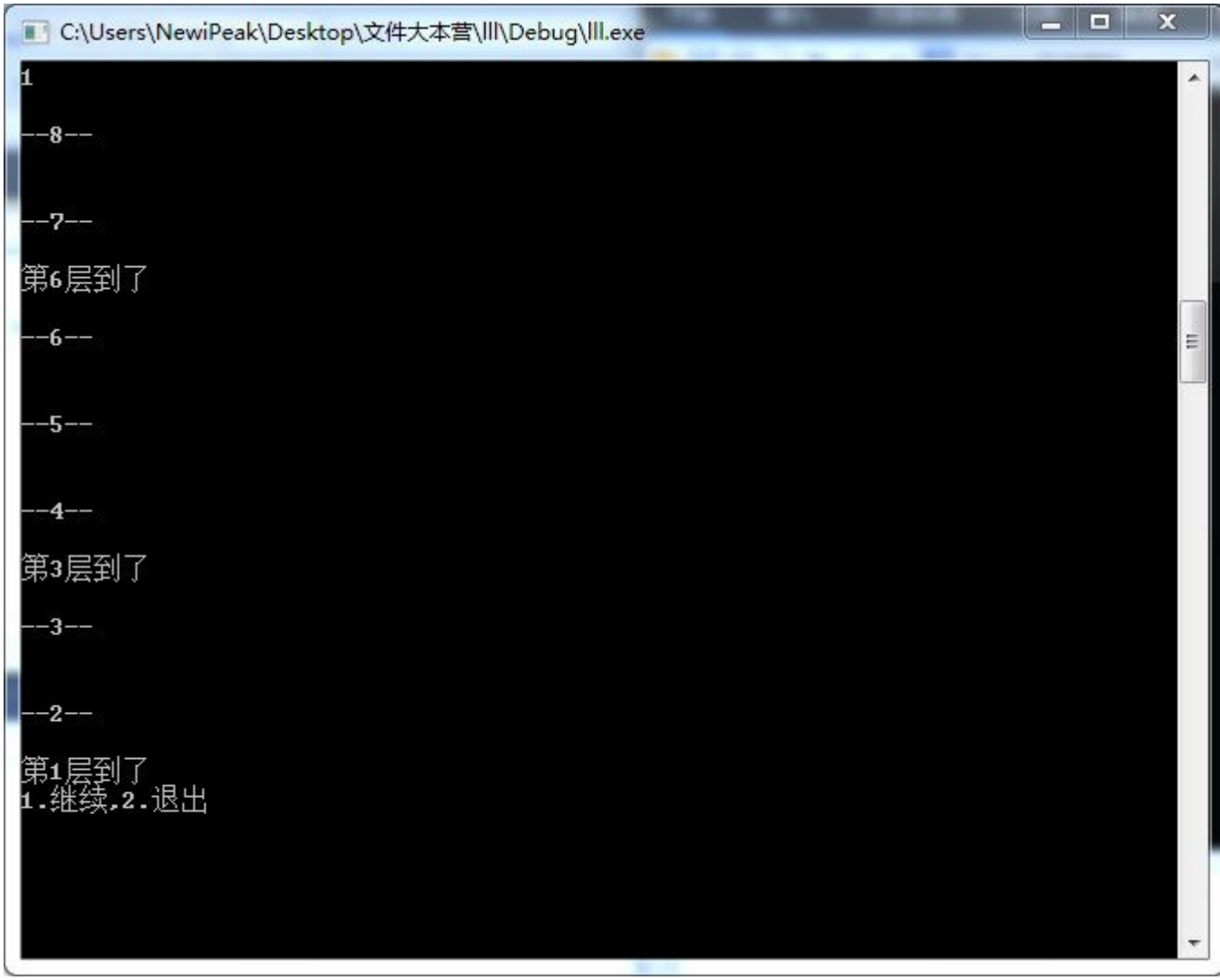
```
                cout<<"1.继续,2.退出"<<endl;
                cin>>k;
             break;}
            }
            }while(k==1);
}
```

六 运行结果


C:\Users\NewiPeak\Desktop\文件大本营\lll\Debug\lll.exe
--3--
第4层到了
--4--
--5--
--6--
--7--
第8层到了
--8--
第9层到了
1.继续,2.退出
1
-----请选择操作-----:
1、上升请按1;2、下降请按2

2
请输入乘坐电梯人数


C:\Users\NewiPeak\Desktop\文件大本营\lll\Debug\lll.exe
1
-----请选择操作-----:
1、上升请按1;2、下降请按2
2
请输入乘坐电梯人数
3
请输入所在的楼层:
8
请输入所在的楼层:
6
请输入所在的楼层:
3
请输入所去的楼层:
6
请输入所去的楼层:
3
请输入所去的楼层:
1
--8--
--7--
第6层到了



七 实验心得

类中用于保存各乘客所在楼层和目的楼层，可以定义一个数组进行保存。

电梯的运行要便捷并符合常理，需要动脑好好思考。