

# 面向对象与 C++ 程序设计

## 第 3 次考试试卷

2020 年 7 月 02 日

### 题目说明

请各位考生从课程信息发布网站下载文本数据文件 `workers.txt` 和源程序文件 `main.cpp`。将上述两个文件保存在 D 盘根目录下。文件 `workers.txt` 中存储了若干钳工的基本信息，一行存储了一位钳工的信息，存储格式如图 1 所示（每行中的各项之间以制表符 `\t` 进行分隔。*斜线文字和线条是用于对文件格式进行解释说明的，在文件中并不存在。*）：

工号	姓名	性别	年龄	钳工级别
1	TOM1WANG	1	48	2
66	HU6	0	56	3
.....				

图 1 数据文件格式示意图

### 基类说明

现有 `CWorker` 类描述如下：

```
class CWorker
{
private:
    int m_ID;          //工号
    string m_Name;     //姓名
    int m_Sex;         //性别    0代表女性    1代表男性
    int m_Age;         //年龄

public:
    CWorker(int ID, string Name, int Sex, int Age)
    {
        if (Sex<0 || Sex>1)
        {
            cout<<"性别初始值"<<Sex<<"不符合要求！"<<endl;
            exit(0);
        }
        if (Age<=0)
        {
            cout<<"年龄初始值"<<Age<<"不符合要求！"<<endl;
            exit(0);
        }
    }
};
```

```

    }

    m_ID=ID;
    m_Name=Name;
    m_Sex=Sex;
    m_Age=Age;
}

string GetName()const
{
    return m_Name;
}

int GetSex()const
{
    return m_Sex;
}

int GetAge()const
{
    return m_Age;
}

friend ostream& operator<<(ostream& Out, const CWorker& Obj)
{
    Out<<right<<setw(5)<<Obj.m_ID;
    Out<<right<<setw(10)<<Obj.m_Name;
    Out<<right<<setw(5)<<Obj.m_Sex;
    Out<<right<<setw(6)<<Obj.m_Age;

    return Out;
}
};

```

## 派生类说明

请以CWorker类为基类进行公有继承，派生出CBenchWorker类用于描述钳工。CBenchWorker类在继承CWorker所有数据成员的基础上，还需要添加1个私有成员用于描述钳工级别，成员定义代码如下：

```
int m_Grade; //钳工级别 1-5，最大为5级
```

- 为 CBenchWorker 类添加一个构造函数。构造函数通过 6 个形式参数对钳工的 6 个成员分别进行初始化。如果这 6 个形式参数都没有提供初始值，则各成员的初始值如下所示：
  - 工号：0
  - 姓名："No Name"
  - 性别：'M'
  - 年龄：18
  - 级别：1
- 为 CBenchWorker 类添加返回级别的成员函数 GetGrade；
- 为 CBenchWorker 类添加前自增运算符++。++运算符用于将钳工的级别加 1；
- 为 CBenchWorker 类重载流输出运算符<<。其中基类部分数据的输出格式与基类的operator<<函数相同，CBenchWorker 类新增成员的输出格式要求如下：
  - 级别：右对齐，占 5 列，不足部分补空格
- 不允许在题目要求以外添加其它成员变量、成员函数或友元函数。

## 程序业务流程说明

- 编写一个函数 **readData**。从下载的数据文件 `workers.txt` 中读取数据，逐个生成 `CBenchWorker` 类的对象，并将这些对象存储到 `vector` 对象 `myVect` 中。
- 编写一个函数 **DisplayData**。显示这些钳工信息。每个钳工数据占一行，每行中显示格式的要求与 `CBenchWorker` 类 `operator<<` 函数的要求相同。
- 编写一个函数 **InecData**。对所有钳工依次执行前自增运算，结果仍然保存在 `myVect` 中。
- 编写一个函数 **DelData**。删除所有年龄大于等于 `X` 岁、级别小于等于 `Y` 级的钳工，结果仍然保存在 `myVect` 中，并返回删除人数。
- 编写一个函数 **DelData**。删除所有姓名中包含（大小写不敏感）字符串 `X` 的钳工，结果仍然保存在 `myVect` 中，并返回删除人数。例如，以目标字符串“Tom”为例，“TOMWANG” “WANGtOM” “WANGtOm” “toMWANG” “ToMWANG” “tomWANG” 中都包含了“Tom”子串。
- 编写一个函数 **SortData**。对所有钳工按照性别（女士优先，女性在前、男性在后）进行排序，其中男性钳工中按照级别进行增序排列，女性钳工中按照级别进行降序排列。
- 编写一个函数 **WriteData**。将经过上述处理后的所有钳工信息保存到指定文件中。每个钳工数据占一行，每行中文件保存格式的要求与 `CBenchWorker` 类 `operator<<` 函数的要求相同。
- 程序运行结果如图 2 所示（[示意图为标准答案](#)）。

```
C:\WINDOWS\system32\cmd.exe
文件读取后一共有 11 条钳工数据:
1  TOM1WANG  1  48  2
2  tomWANG   1  51  2
3  TOmmWANG  1  20  2
4  TOMLI     1  22  3
60  HU7       1  56  1
62  HU2       1  52  2
63  HU3       1  53  1
61  HU1       1  51  3
64  HU4       0  54  1
65  HU5       0  55  2
66  HU6       0  56  3

第1次删除了1个钳工数据
第2次删除了4个钳工数据
上述流程结束后一共有 6 条钳工数据:
66  HU6       0  56  4
65  HU5       0  55  3
64  HU4       0  54  2
63  HU3       1  53  2
62  HU2       1  52  3
61  HU1       1  51  4

文件保存成功
请按任意键继续. . .
```

图 2 程序运行结果示意图

## 编程要求

- 根据业务流程和 main.cpp 中的注释，在 main.cpp 的基础上将程序补充完整。
- 不可以修改 CWorker 类的代码。每修改一处编程题总得分扣 2 分，最多扣 10 分。
- 不允许在题目要求以外为 CBenchWorker 类添加其它成员变量、成员函数或友元函数。。每修改一处编程题总得分扣 2 分，最多扣 10 分。
- 不可以修改 main 函数的代码。每修改一处编程题总得分扣 2 分，最多扣 10 分。

## 评分标准

（编程题满分为 100 分）

大项	子项	评分项	应得分	实得分
正确性 90 分	结果（90 分）	CBenchWorker 类结构	5	
		CBenchWorker 类构造函数	5	
		CBenchWorker 类获取专业和级别函数	5	
		CBenchWorker 类前自增	10	
		CBenchWorker 类流输出	10	
		ReadData 函数	10	
		DisplayData 函数	5	
		InecData 函数	5	
		DelData 函数 1（int 版本）	10	
		DelData 函数 2（string 版本）	10	
		SortData 函数	10	
		WriteData 函数	5	
	异常情况	程序运行出现异常（运行错、死循环等）	-10	
		修改 main 函数	每处-2/最多-10 分	
		修改基类接口	每处-2/最多-10 分	
		修改子类接口	每处-2/最多-10 分	
		编译、连接不通过	以上合计为 0 分	
可读性 10 分	缩进对齐（4 分）	正确运用缩进对齐规则	4	
		有缩进对齐但不完全符合要求	2	
		没有使用缩进对齐规则	0	
	注释（4 分）	有详细且正确的注释	4	
		有注释，但不够详细	2	
		完全没有注释	0	
	变量命名（2 分）	变量命名有规则	2	
		变量命名有规则、但规则使用不一致	1	
		变量命名无规则	0	

