

# 实验四 对象作为数据成员

## 1 实验目的

学习对象作为类的数据成员的使用方法，包括以下几个方面：

- (1) `const` 对象作为类的数据成员；
- (2) 对象数据成员的初始化以及“成员初始化器”的使用方法；
- (3) `const` 成员函数的声明、定义及使用。

## 2 实验内容

### 2.1 使用 `Date` 类，定义 `Employee` 类（50 分）

#### (1) 问题描述

在《实验三 面向对象初步》中，设计了日期类 `Date`。本次实验将设计雇员类 `Employee`，并将日期类 `Date` 作为雇员类 `Employee` 的内嵌数据成员。日期类 `Date` 和雇员类 `Employee` 的声明分别如下所示：

```
class Date{
public:
    /* 默认构造函数，以fullyear的形式给出年月日，默认值为1990年1月1日，同时设置
    日期分隔符为“-” */
    Date(int year = 1990, int month = 1, int day = 1);

    /* get、set方法 */
    // 设置日期，如果有非法的月或日，将其置为1
    void setDate(int year, int month, int day);
    void setYear(int year);
    int getYear() const;
    void setMonth(int month);
    int getMonth() const;
    void setDay(int month);
    int getDay() const;
    void setSeparator(char separator);
```

```

/* 输出函数，请使用setfill('0')和setw(2)。*/
void printFullYear() const; //以YYYY-MM-DD的形式打印，2011-01-08
void printStandardYear() const; //以YY-MM-DD的形式打印，比如11-01-08
/* 计算当前日期与参数日期之间相差几个整年，仅考虑参数日期比当前日期晚的情况。
注意参数为日期对象的引用。*/
int fullYearsTo(const Date &date) const;
/* 计算当前日期与参数日期之间相差多少天(考虑闰年)，如果参数日期在当前日期之前，
返回负数。注意参数为日期对象的引用。*/
int daysTo(const Date &date) const;
/* 新增函数，可以被daysTo函数调用 */
int getDayOfYear() const; //计算当前日期是本年的第几天
int getLeftDaysYear() const; //计算当前日期距本年结束还有几天，不包括当前日期这一天
private:
    int year;
    int month;
    int day;
    char separator; // 日期分隔符
/* 新增数据成员和函数成员 */
/*声明静态常量变量，每月的天数，在.cpp文件中定义并初始化 */
static const int DAYS_PER_MONTH[12];
/*根据年和月，判断参数日期是否合法。如果合法，返回day，否则返回1。*/
int checkDay(int day) const;
bool isLeapyear(int year) const; //断参数年是否是闰年。
};

class Employee{
public:
    //构造函数，使用“成员初始化器”初始化数据成员
    Employee(const string, const string, const Date&, const Date&);
    //打印员工的信息。调用Date类的print函数，打印员工的生日和雇佣日期。
    void print() const;
    //计算员工在参数指定的日期时，满多少岁。请使用Date类的fullYearsTo函数
    int getAge(Date& date) const;
    //计算该员工在参数指定的日期时，工作满了多少年。
    int getYearsWorked(Date& date) const;

```

```

//计算该员工在参数指定的日期时，工作了多少天。使用Date类的daysTo函数。
int getDaysWorked(Date& date) const;
~Employee();    //析构函数
private:
    string firstName;
    string lastName;
    /* 要求：出生日期、雇用日期必须声明为const，因为对于每个雇员，这两个属性都是
    不会变化的。*/
    const Date birthDate;    //内嵌对象，出生日期
    const Date hireDate;    //内嵌对象，雇用日期
};

```

## (2) 问题要求

可以满足以下主函数的要求：

```

#include <iostream>
#include <string>
using namespace std;
void main() {
    Date birth(1969, 8, 11);
    Date hire(1998, 4, 1);
    Date today(2010, 4, 30);
    Employee manager("Bob", "Blue", birth, hire);
    cout << endl;
    manager.print();
    cout << endl;
    cout << manager.getAge(today) << endl;
    cout << manager.getDaysWorked(today) << endl;
}

```

输出结果如下，其中的注释仅为了说明运行结果，实际执行时不必输出。

```

//调用manager.print()后，打印以下信息
Blue, Bob Hired: 1998-04-01 Birthday: 1969-08-11
//调用manager.getAge(today)后，打印以下信息
40 // 工作满了40年
//调用manager.getDaysWorked(today)
4412 // 已工作了4412天

```

## 2.2 设计一个 CD 播放机 CDPlayer (50 分)

### (1) 问题描述

设计一个CD播放机CDPlayer,它能够播放CD中的歌。其中,CD类、CDPlayer类的声明分别如下所示:

```
class CD {
public:
    CD(string name, string songs[]);
    string getSonger() const;    // 获得歌手的名称
    string getSong(int index) const; // 获得某首歌的歌名
    void listSongs() const;    // 列出CD的内容
private:
    string songer;    // 歌手的名字。
    string songs[6];    // 每张专辑6首歌的名字。
};

class CDPlayer {
public:
    CDPlayer();

    /*提供给用户一个按钮菜单,通过这个菜单,用户可以选择:
    1. 插入CD
    2. 播放CD
    3. 弹出CD
    0. 关机    */
    void showButtons() const;

    /*插入CD. void insertCD(CD* cd), 形参是指向CD对象的指针。如果CDPlayer
    中已经有CD, 提示先取出CD;如果CDPlayer中没有CD, 显示插入了哪位歌星的CD。*/
    void insertCD(CD *cd);

    /*弹出CD. CD* ejectCD(), 返回值是指向该CD对象的指针。如果CDPlayer中已经
    有CD, 显示弹出了哪位歌星的CD, 返回该CD的指针;如果CDPlayer中没有CD, 提示
    CDPlayer中没有CD, 返回NULL。*/
    CD *ejectCD();
```

```

/*播放CD。如果CDPlayer中已经有CD，显示正在播放哪位歌星的CD，并打印CD中歌曲
的清单；如果CDPlayer中没有CD，显示CDPlayer中没有CD，并提示用户插入CD。*/
void play() const;
private:
    /* 插入CDPlayer中的CD，它是指向CD对象的指针。没有CD时，为null。使用指针，
    好地模拟了CD对象不是播放器的一部分，播放器只是读取放入其中的CD的内容。 */
    CD *cd;
    bool cdIn; // CDPlayer中是否已经插入CD
};

```

### 主函数如下：

```

#include <iostream>
#include <string>
using namespace std;

void main() {
    string name;
    string songs[6];

    cout << "\n制造CD.....\n";
    // 输入歌手名字
    cout << "  Songer's Name: ";
    cin >> name; // 输入：周杰伦

    // 输入该歌手的六首歌名（青花瓷、菊花台、三节棍等）
    for (int i = 0; i < 6; i++) {
        cout << "  song" << (i+1) << "#: ";
        cin >> songs[i];
    }

    CD cd(name, songs); //制造CD
    cd.listSongs();      //显示CD的内容
    CDPlayer player;     //制造CDplayer
    player.showButtons();//生成播放机的按钮
}

```

```

/* 播放 */
player.play();          //打印: Please insert CD first
/* 插入CD */
player.insertCD(&cd);    //打印: 插入了周杰伦的CD.....
/* 播放 */
player.play();          //打印: 正在播放周杰伦的CD.....
player.ejectCD();        //打印: 弹出了周杰伦的CD.....

/* 另造一张CD, 歌手和歌的录入省略。*/
CD cd2(name2, songs2);
player.insertCD(cd2);
player.play();
}

```

## (2) 问题要求

程序的执行结果如下:

制造 CD.....

Songer's Name: 周杰伦	}	用户输入
song1#: 青花瓷		
song2#: 菊花台		
song3#: 三节棍		
song4#: 东风破		
song5#: 珊瑚海		
song6#: 稻香		

Songer: 周杰伦	}	cd.listSongs() 的执行结果
1. 青花瓷		
2. 菊花台		
3. 三节棍		
4. 东风破		
5. 珊瑚海		
6. 稻香		

*****	}	player.showButtons() 的执行结果
* 1. 播放 CD *		
* 2. 插入 CD *		
* 3. 弹出 CD *		
* 0. 关机 *		
*****		

Please insert CD first (player.play() 的执行结果)

插入了周杰伦的 CD..... (player.insertCD(cd) 的执行结果)

正在播放周杰伦的 CD.....

Songer: 周杰伦

1. 青花瓷
2. 菊花台
3. 三节棍
4. 东风破
5. 珊瑚海
6. 稻香



player.play() 的执行结果

弹出了周杰伦的 CD.....

(player.ejectCD() 的执行结果)