上机实验报告 2

姓名：王越洋　　学号：22009200894

一、实验一

（1）题目

编写设计一个 People（人）类。该类的数据成员有年龄（age）、身高（height）、体重（weight）和人数（num），其中人数为静态数据成员，成员函数有构造函数（People）、进食（Eating）、运动（Sporting）、睡眠（Sleeping）、显示（Show）和显示人数（ShowNum）。

其中构造函数由已知参数年龄(a)、身高(h)和体重(w)构造对象，进食函数使体重加 1，运动函数使身高加 1，睡眠函数使年龄、身高、体重各加 1，显示函数用于显示人的年龄、身高、体重，显示人数函数为静态成员函数，用于显示人的个数。假设年龄的单位为岁，身高的单位为厘米，体重的单位为市斤，要求所有数据成员为 protected 访问权限，所有成员函数为 public 访问权限，在主函数中通过对象直接访问类的所有成员函数。

（2）代码：

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

class People {
protected:
    int age;
    int height;
    int weight;
    static int num;

public:
    People(int a, int h, int w) : age(a), height(h), weight(w) {
        num++;
    }

    void Eating() {
        weight += 1;
    }

    void Sporting() {
        height += 1;
    }

    void Sleeping() {
        age += 1;
```

```
        height += 1;
        weight += 1;
    }

    void Show() const {
        cout << "Age: " << age << " years, Height: " << height << " cm
   , Weight: " << weight << " kg" << endl;
    }

    static void ShowNum() {
        cout << "Number of people: " << num << endl;
    }

    ~People() {
        num--;
    }
};

int People::num = 0;

int main() {
    People p1(25, 170, 60);
    p1.Show();
    p1.Eating();
    p1.Show();
    People::ShowNum();

    People p2(30, 180, 70);
    p2.Show();
    People::ShowNum();

    return 0;
}
```

（2）代码分析

1. 数据成员：age（年龄）、height（身高）、weight（体重）为对象的基本属性，num 是静态数据成员，用于记录创建的对象总数（人数）。num 为静态变量，属于整个类而非某个特定对象。

2. 构造函数：构造函数用来初始化对象的属性，并在每创建一个对象时增加人数（num++）。

3. 成员函数：

Eating()：表示进食，体重加 1。

Sporting()：表示运动，身高加 1。

Sleeping()：表示睡眠，年龄、身高、体重各加 1。

Show()：用于显示个人的年龄、身高、体重信息。

ShowNum()：静态成员函数，显示当前创建的 People 对象的总数（人数）。

4.析构函数：当一个对象被销毁时，人数减少（num--）。

5.静态成员函数：可以直接通过类名调用，不依赖于某个对象，且只能访问静态数据成员。

二、实验二

（1）题目

定义一个描述学生(Student)基本情况的类,数据成员包括姓名(name)、学号(num)、数学成绩(mathScore)、英语成绩(englishScore)、人数(count)、数学总成绩(mathTotalScore)和英语总成绩(englishTotalScore)。

其中姓名定义为长度为 18 的字符数组，其他数据成员类型为整型，数学总成绩、英语总成绩和人数为静态数据成员，函数成员包括构造函数、显示基本数据函数（ShowBase）和显示静态数据函数(showStatic)，其中构造函数由已知参数姓名(nm)、学号(nu)、数学成绩(math)和英语成绩(english)构造对象，显示基本数据函数用于显示学生的姓名、学号、数学成绩、英语成绩，显示静态数据函数为静态成员函数，用于显示人数、数学总成绩、英语总成绩；

要求所有数据成员为 private 访问权限，所有成员函数为 public 访问权限，在主函数中定义若干个学生对象，分别显示学生基本信息，以及显示学生人数，数学总成绩与英语总成绩。

（2）代码

```
1. #include <iostream>
2. #include <cstring>
3. using namespace std;
4.
5. class Student {
6. private:
7.     char name[18];
8.     int num;
9.     int mathScore;
10.    int englishScore;
11.    static int count;
12.    static int mathTotalScore;
13.    static int englishTotalScore;
14.
15.public:
16.    Student(const char* nm, int nu, int math, int english) {
17.        strncpy(name, nm, 18);
18.        num = nu;
19.        mathScore = math;
20.        englishScore = english;
21.        count++;
22.        mathTotalScore += math;
23.        englishTotalScore += english;
```

```
    }

    void ShowBase() const {
        cout << "Name: " << name << ", ID: " << num
             << ", Math Score: " << mathScore
             << ", English Score: " << englishScore << endl;
    }

    static void showStatic() {
        cout << "Total Students: " << count
             << ", Total Math Score: " << mathTotalScore
             << ", Total English Score: " << englishTotalScore << end
   l;
    }

    ~Student() {
        count--;
        mathTotalScore -= mathScore;
        englishTotalScore -= englishScore;
    }
};

int Student::count = 0;
int Student::mathTotalScore = 0;
int Student::englishTotalScore = 0;

int main() {
    Student s1("Alice", 1001, 90, 85);
    Student s2("Bob", 1002, 80, 88);

    s1.ShowBase();
    s2.ShowBase();
    Student::showStatic();

    return 0;
}
```

（3）代码分析

1. 数据成员：name（姓名）、num（学号）、mathScore（数学成绩）、englishScore（英语成绩）为私有成员，保存每个学生的基本信息。count、mathTotalScore、englishTotalScore 是静态数据成员，记录学生的总人数以及数学、英语的总成绩。

2. 构造函数：使用构造函数初始化学生对象，并累加静态成员变量 count、mathTotalScore 和 englishTotalScore，以便跟踪创建的学生数量和总成绩。

3. 成员函数：

ShowBase()：显示学生的基本信息，包括姓名、学号、数学成绩、英语成绩。
showStatic()：静态成员函数，显示总人数和数学、英语的总成绩，可以直接通过类名调用。

4.析构函数：当对象被销毁时，人数减少，同时从总成绩中减去该对象的数学和英语成绩。

5.静态成员变量：count 和总成绩的累加使得可以跟踪所有学生的数量和成绩，而不是局限于单个学生对象。

三、实验三

（1）题目

定义一个 Dog，包含 name、age、sex 和 weight 等属性以及对这些属性操作的方法。要求用字符指针描述 name，并且用对象指针来测试这个类。

（2）代码

```
#include <iostream>
#include <cstring>
using namespace std;

class Dog {
public:
        char* name;
        int age;
        char sex;
        float weight;

        Dog(const char* n, int a, char s, float w) {
                name = new char[strlen(n) + 1];
                strcpy(name, n);
                age = a;
                sex = s;
                weight = w;
        }

        void Show() const {
                cout << "Name: " << name << ", Age: " << age
                        << ", Sex: " << sex << ", Weight: " << weight << " kg" << endl;
        }

        ~Dog() {
                delete[] name;
        }
};

```

```
int main() {
        Dog* d1 = new Dog("Buddy", 3, 'M', 10.5);
        d1->Show();
        delete d1;

        return 0;
}

```

（3）代码分析

1.数据成员：name（名字）、age（年龄）、sex（性别）、weight（体重）是公共成员，描述狗的基本信息。名字使用动态内存分配，采用字符指针存储。

2.构造函数：接受参数来初始化狗的名字、年龄、性别和体重。name 需要动态分配内存，并复制传入的名字。

3.成员函数：

Show()：显示狗的名字、年龄、性别和体重信息。

4.析构函数：析构函数释放动态分配的 name 的内存，防止内存泄漏。

5.对象指针测试：通过对象指针创建和操作 Dog 类实例，测试动态内存管理和指针的正确性。

四、实验四

（1）题目

管理个人活期账户：个人储蓄活期账户包括账号、户名、密码、余额、活期年利率等信息。要求能够对个人账户进行存钱、取钱、计算年利息、打印账户相关信息等操作。编写主函数测试账户相关功能。

（2）代码

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

class Account {
private:
    int accountNumber;      // 账号
    string accountName;     // 户名
    string password;        // 密码
    double balance;         // 余额
    static double rate; // 活期年利率

public:
    Account(int accNum, string accName, string pwd, double bal)
        : accountNumber(accNum), accountName(accName), password(pwd), balance(bal) {}

    void Deposit(double amount) {
        balance += amount;
```

```
18.         cout << "存入金额: " << amount << " 元, 当前余
    额: " << balance << " 元" << endl;
19.     }
20.
21.     void Withdraw(double amount) {
22.         if (amount > balance) {
23.             cout << "余额不足，无法取出该金额。" << endl;
24.         } else {
25.             balance -= amount;
26.             cout << "取出金额: " << amount << " 元, 当前余
    额: " << balance << " 元" << endl;
27.         }
28.     }
29.
30.     void Calculate() const {
31.         double interest = balance * rate;
32.         cout << "本年度利息: " << interest << " 元" << endl;
33.     }
34.
35.     void ShowAccountInfo() const {
36.         cout << "账号: " << accountNumber
37.              << ", 户名: " << accountName
38.              << ", 余额: " << balance << " 元" << endl;
39.     }
40.};
41.
42.double Account::rate = 0.02;
43.
44.int main() {
45.     Account acc1(123456, "张三", "password", 1000.0);
46.     acc1.ShowAccountInfo();
47.     acc1.Deposit(500);
48.     acc1.Withdraw(300);
49.     acc1.Calculate();
50.
51.     return 0;
52.}
```

（3）代码分析

1. 数据成员：accountNumber（账号）、accountName（户名）、password（密码）、balance（余额）为每个账户的属性。静态成员 interestRate 用于表示所有账户共享的活期年利率。

2. 构造函数：接受账户信息（账号、户名、密码和初始余额）来初始化对象。interestRate 是静态的，不随对象变化。

3.成员函数：

Deposit()：接受存入金额，更新账户余额。

Withdraw()：检查账户余额，若足够则取出指定金额并更新余额，否则提示余额不足。

CalculateInterest()：根据当前余额和年利率计算账户的年利息。

ShowAccountInfo()：显示账户的基本信息，如账号、户名和余额。

4.静态成员变量：interestRate 是活期年利率，所有账户共享，通过类名直接访问，不依赖于具体对象。