实验五、函数实验

计算机学院 熊明 20305055

一、实验目的

1. 了解自顶向下TOP-DOWN结构化程序设计方法（面向过程编程）
2. 掌握函数的定义、声明的方法；
3. 掌握函数的编写要求；
4. 掌握函数的调用方法；
5. 掌握函数参数的传递方法；
6. .掌握多文件编程方法。

二、实验原理

将一些功能封装成函数，在main函数中通过调用函数的方法实现功能

三、实验内容

1. 函数版本的计算器

```
#include<iostream>
using namespace std;
void add(int x,int y){
    cout<<x<<"+"<<y<<"="<<x+y<<endl;
}
void delet(int x,int y){
    cout<<x<<"-"<<y<<"="<<x-y<<endl;
}
void mul(int x,int y){
    cout<<x<<"*"<<y<<"="<<x*y<<endl;
}
void divide(int x,int y){
    cout<<x<<"/"<<y<<"="<<x/y<<endl;
}
int main(){
    double x,y;
    int chose,isFlag = 1;
    do{
        cout<<"*********计算器*********"<<endl;
        cout<<"*********1.加法*********"<<endl;
        cout<<"*********2.减法*********"<<endl;
        cout<<"*********3.乘法*********"<<endl;
        cout<<"*********4.除法*********"<<endl;
        cout<<"*********5.退出*********"<<endl;
        cout<<"*******请选择1-5：*******"<<endl;
        cin>>chose;
        if(chose>5||chose<1){
            cout<<"非法输入，请重试"<<endl;
        }else{
            if(chose==5){
```

```
                isFlag=0;
                continue;
            }
            cout<<"输入x的值为："<<endl;
            cin>>x;
            cout<<"输入y的值为："<<endl;
            cin>>y;
            switch (chose)
            {
            case 1:
                add(x,y);
                break;
            case 2:
                delet(x,y);
                break;
            case 3:
                mul(x,y);
                break;
            case 4:
                divide(x,y);
                break;
            default:
                break;
            }
        }
    } while(isFlag);
    cout<<"程序退出~"<<endl;
}
```

2. 研读下面代码，改写成函数实现

```
#include<iostream>
using namespace std;
int numa_to_numb(int a,int b){
    int ans = 1;
    for(int i;i<b;i++){
        ans *= a;
    }
    return ans;
}
int main(){
    cout<<"3^4 is "<<numa_to_numb(3,4)<<endl;
    cout<<"6^5 is "<<numa_to_numb(6,5)<<endl;
    cout<<"12^10 is "<<numa_to_numb(12,10)<<endl;
}
```

发现结果错误，经过debug后发现是循环忘记将i赋值为0，改进后代码为：

```
#include<iostream>
using namespace std;
int numa_to_numb(int a,int b){
    int ans = 1;
    for(int i=0;i<b;i++){
        ans *= a;
```

```
    }
    return ans;
}
int main(){
    cout<<"3^4 is "<<numa_to_numb(3,4)<<endl;
    cout<<"6^5 is "<<numa_to_numb(6,5)<<endl;
    cout<<"12^10 is "<<numa_to_numb(12,10)<<endl;
}
```

运行结果：

```
PS E:\vscode>  & 'c:\Users\15989\.vscode\extensions\ms-vscode.cp
icrosoft-MIEngine-In-ww0fdv3z.stl' '--stdout=Microsoft-MIEngine-
osoft-MIEngine-Pid-ouqvtnic.5bc' '--dbgExe=C:\MinGW\mingw64\bin\
3^4 is 81
6^5 is 7776
12^10 is 1787822080
```

numa_to_numb函数实现了任意整数a的b次幂运算。

3. 自定义函数尝试使图案旋转起来：

```
#include <graphics.h>
#include<conio.h>
#include<iostream>
#define PI 3.14159
using namespace std;
int tangel = 20;
double B = tangel * PI / 180;
void rotate1(double &a,double &b) {
    double x = a, y = b;
    a = x * cos(B) - y * sin(B) - 160 * cos(B) + 240 * sin(B) + 160;
    b = x * sin(B) + y * cos(B) - 160 * sin(B) - 240 * cos(B) + 240;
}
void rotate2(double& a, double& b) {
    double x = a, y = b;
    a = x * cos(B) - y * sin(B) - 500 * cos(B) + 240 * sin(B) + 500;
    b = x * sin(B) + y * cos(B) - 500 * sin(B) - 240 * cos(B) + 240;
}
int main(){
    initgraph(640, 480);
    double a1 = 70, a2 = 160, a3 = 250, a4 = 160, b1 = 150, b2 = 240, b3 = 330, b4 = 240;
    double aa1 = 600, aa2 = 400, aa3 = 500, aa4 = 500, bb1 = 240, bb2 = 240, bb3 = 340, bb4 = 140;
    while (1) {

        fillcircle(160, 240, 25);//中心圆

        fillcircle(a1, b4, 50);//
        rotate1(a1, b4);
        fillcircle(a3, b2, 50);
        rotate1(a3, b2);
        fillcircle(a2, b3, 50);
        rotate1(a2, b3);
        fillcircle(a4, b1, 50);
```

```
        rotate1(a4, b1);

        fillcircle(500, 240, 50);//中心圆

        fillcircle(aa1, bb1, 25);
        rotate2(aa1, bb1);
        fillcircle(aa2, bb2, 25);
        rotate2(aa2, bb2);
        fillcircle(aa3, bb3, 25);
        rotate2(aa3, bb3);
        fillcircle(aa4, bb4, 25);
        rotate2(aa4, bb4);
        Sleep(150);
        cleardevice();
    }
    closegraph();
    return 0;
}
```

参考图像(点)绕任意中心位置旋转的旋转矩阵推导

设置颜色，参考setColor方法：设置颜色

```
#include <graphics.h>
#include<conio.h>
#include<iostream>
#define PI 3.14159
using namespace std;
int tangel = 20;
double B = tangel * PI / 180;
void rotate1(double &a,double &b) {
    double x = a, y = b;
    a = x * cos(B) - y * sin(B) - 160 * cos(B) + 240 * sin(B) + 160;
    b = x * sin(B) + y * cos(B) - 160 * sin(B) - 240 * cos(B) + 240;
}
void rotate2(double& a, double& b) {
    double x = a, y = b;
    a = x * cos(B) - y * sin(B) - 500 * cos(B) + 240 * sin(B) + 500;
    b = x * sin(B) + y * cos(B) - 500 * sin(B) - 240 * cos(B) + 240;
}
int main(){
    initgraph(640, 480);
    double a1 = 70, a2 = 160, a3 = 250, a4 = 160, b1 = 150, b2 = 240, b3 = 330, b4 = 240;
    double aa1 = 600, aa2 = 400, aa3 = 500, aa4 = 500, bb1 = 240, bb2 = 240, bb3 = 340, bb4 = 140;
    while (1) {
        setfillcolor(YELLOW);//设置填充圆的颜色
        fillcircle(160, 240, 25);//中心圆
        setfillcolor(RED);//设置填充圆的颜色
        fillcircle(a1, b4, 50);//
        rotate1(a1, b4);
        setfillcolor(BLUE);//设置填充圆的颜色
        fillcircle(a3, b2, 50);
        rotate1(a3, b2);
```

```
        setfillcolor(GREEN);//设置填充圆的颜色
        fillcircle(a2, b3, 50);
        rotate1(a2, b3);
        setfillcolor(MAGENTA);//设置填充圆的颜色
        fillcircle(a4, b1, 50);
        rotate1(a4, b1);

        setfillcolor(CYAN);//设置填充圆的颜色
        fillcircle(500, 240, 50);//中心圆

        setfillcolor(MAGENTA);//设置填充圆的颜色
        fillcircle(aa1, bb1, 25);
        rotate2(aa1, bb1);
        setfillcolor(BLUE);//设置填充圆的颜色
        fillcircle(aa2, bb2, 25);
        rotate2(aa2, bb2);
        setfillcolor(GREEN);//设置填充圆的颜色
        fillcircle(aa3, bb3, 25);
        rotate2(aa3, bb3);
        setfillcolor(YELLOW);//设置填充圆的颜色
        fillcircle(aa4, bb4, 25);
        rotate2(aa4, bb4);
        Sleep(100);
        cleardevice();
    }
    closegraph();
    return 0;

```

四、实验心得体会

debug是很好的代码自我检测工具，在编程大工程的时候，往往很难发现自己犯的小错误，这个时候debug工具就非常重要了。