实验三 程序控制结构

计算机学院 熊明 20305055

一、实验目的

- 1. 了解 C ++语言表示逻辑量的方法 (以 0 代表"假",以非 0 代表"真")。
- 2. 掌握关系表达式和逻辑表达式的使用。
- 3. 熟练使用if语句进行程序设计。
- 4. 结合程序掌握一些简单的算法。
- 5. 学会使用选择结构解决一般的实际问题, 能编写简单的应用程序。
- 6. 理解结构化程序设计方法的循环结构
- 7. 掌握用 while 语句实现循环结构
- 8. 掌握用嵌套while语句实现循环结构
- 9. 使用循环来解决问题
- 10. 通过循环语句初步了解算法复杂度

二、实验原理

利用顺序、循环、分支结构解决问题。

三、实验内容

1. 实现乘法表

打印如下九九乘法表。 要求至少用两种循环语句实现。改进乘法口诀表的显示模式 (完整, 左上, 左下, 右上, 右下) 、

代码如下:

运行结果如下:

```
★ 文件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 运行(R) 终端(T) 帮助(H)
                                                                                                                                                                                                                                                                                 运行和... ▷ JavaScr∨ 鬱 ··· м Os_lab2.asm • lab2.py 2 • lab2_1.cpp ×
                                                                                                                                                                                                                                                                                                                 $>∨ 🕸 🖽 …
                                                                           chengshe2_exp > lab2 > C lab2_1.cpp > .
2    using namespace std;
                变量
                                                                                        int main()
                                                                                                  for(int i=1;i<10;i++){
                                                                                                          for(int j=1;j<10;j++){
    cout<<i<<"*"<<j<<"="<<i*j<<' ';
 ź
                                                                                                          cout<<endl:
                                                                               ···cout<<endl;
                                                                             问题 2 输出 调试控制台 终端
                                                                             bugLauncher.exe' '--stdin=Microsoft-MIEngine-In-Syfmw4ad.aiy' '--stdout=Microsoft-MIEngine-Out-olq12ggo.fym' '--
tderr=Microsoft-MIEngine-Error-jrlgoog0.wce' '--pid=Microsoft-MIEngine-Pid-lctgwbf3.k5j' '--dbgExe=C:\MinGW\mingw
4\bin\gdb.exe' '---interpreter=mi'
11-1 1*2-2 1*3-3 1*4-4 1*5-5 1*6-6 1*7-7 1*8-8 1*9-9
2*1-2 2*2-4 2*3-6 2*4-8 2*5-10 2*6-12 2*7-14 2*8-16 2*9-18
3*1-3 3*2-6 3*3-9 3*4-12 3*5-15 3*6-18 3*7-21 3*8-24 3*9-27
4*1-4 4*2-8 4*3-12 4*4-16 4*5-20 4*6-24 4*7-28 4*8-32 4*9-35
5*1-5 5*2-16 5*3-15 5*4-20 5*5-25 5*6-30 5*7-35 5*8-40 5*9-45
6*1-6 6*2-12 6*3-18 6*4-24 6*5-30 6*6-36 6*7-42 6*8-48 6*9-54
7*1-7 7*2-14 7*3-21 7*4-28 7*5-35 7*6-42 7*7-49 7*8-56 7*9-63
8*1-8 8*2-16 8*3-24 8*4-32 8*5-40 8*6-48 8*7-56 8*8-64 8*9-72
9*1-9 9*2-18 9*3-27 9*4-36 9*5-45 9*6-54 9*7-63 9*8-72 9*9-81
9*E:\yscotos\[
                                                                                                                                                                                                                                                                                                                   Python

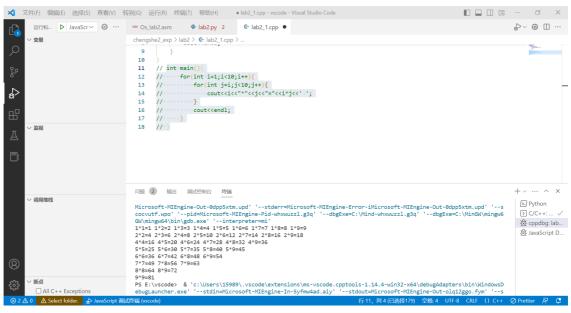
    ∇/C++:..

                                                                                                                                                                                                                                                                                                                     Ø cppdbg: lab...
```

实现左上乘法表,代码如下:

```
#include<iostream>
 2
    using namespace std;
 3
    int main(){
 4
        for(int i=1;i<10;i++){
            for(int j=i;j<10;j++){//小循环从i开始,删去重复的
                cout<<i<<"*"<<j<<"="<<ii*j<<' ';
 6
 7
 8
            cout<<end1;
9
        }
10
   }
```

运行结果如下:



2. 模拟小球的自由落体

ppt上给出的代码位移增量是y+=1,但真正的位移增量不应该是一个常数,故我利用高中所学的物理知识,重新完善了一遍代码。

v1:

```
1 #include<graphics.h>
```

```
2 #include<conio.h>
 3
    #include<stdio.h>
 4
    #include<math.h>
 5
    int main() {
 6
        double y = 50, v1=0, t=0, v2=0, flag=1;
 7
        //g=0.01
 8
        initgraph(600, 600);
 9
        while (1) {
10
            //y += 1;
11
            //未发生碰撞
12
            if (y \ge 20 \& y \le 580) {
13
                 if (flag == 1) {
14
                     //正常下落
15
                     y += 0.05 * t * t;
16
                     v1 = t;
17
                     t++;
                 }
18
19
                 else {
20
                     //减速上升
21
                     y = 0.05 * t * t;
22
                     v2 = t;
23
                     t++;
24
                     if (fabs(v2 - v1) \le 1)  {
25
                         v1 = 0;
                         v2 = 0;
26
27
                         flag = 1;
28
                         t = 0;
29
                     }
30
                 }
31
            }
32
            else if (y >= 580) {
33
                //碰撞
34
                 y = 580;
35
                 v1 = v1 / 2;
36
                 flag = 0;
37
                 t = 0;
38
            }
39
            cleardevice();
40
            fillcircle(300, y, 20);
41
            sleep(10);
42
        }
43
        //_getch();
44
        closegraph();
45
```

第一版代码完成了速度衰减和小球的上下摆动,最终停下来的功能,但随即检查发现小球的减速上升还是有问题,并且如果将碰撞衰减的比例从1/2变成1/1.05,小球会一直卡在一个地方上下摆动,说明代码不够完美,于是进一步改动。

最终代码如下:

```
#include<graphics.h>
#include<conio.h>
#include<stdio.h>
#include<math.h>
```

```
5 #define g 1//令重力加速度为1
6
   int main() {
7
       //假设空气阻力为0,碰撞有速度损失,损失比例可更改
8
       int flag = 1;
9
       double y = 20, v = 0, t = 1, y0 = 20;
10
       initgraph(600, 600);
       while (1) {
11
           if (y > 580) {//触底以后碰撞
12
13
               flag = 0; // 修改标志位, 将其置零, 下面的分支进入<math>else分支
14
               v = -v / 1.05; // 碰撞损失的比例
15
               t = 0;//时间清零
               y = 580;//将球置于底部
16
17
               y0 = 580;
18
           if (y \le 580 \& y \ge 20) {
19
20
               if (flag == 1) {//第一次下落是正常自由落体
21
                  y = y0 + 0.5 * g * t * t;
22
                  v = g * t;
23
                  t++;
24
               }
               else {//包括减速上升和加速下落的方程
25
                  y = y0 + v * t + 0.5 * g * t * t;
26
27
                  t++;
               }
28
29
30
31
           //if (fabs(v - 0) \le 0.001) break;
32
           cleardevice();
33
           fillcircle(300, y, 20);
34
           Sleep(10);
35
       }
36
       //_getch();
37
       closegraph();
38 }
```

四、实验心得体会

实验熟练运用高中物理知识以及分支语句,在不同条件下有不同的结果。