西安交通大学实验报告

课程 计算机程序设计 实验名称 类的构造与析构 第 1 页 共 25 页

系 别 钱 学 森 学 院 实验日期 2017年 5 月 4 日

专业班级 钱62 组别\_\_\_\_\_\_\_\_\_\_\_ 实验报告日期 2017 年 5 月8日

姓 名 周宇晨 学号 2160405046 报 告 退 发 ( 订正 、 重做 )

同 组 人\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 教 师 审 批 签 字

一、实验目的

1.进一步熟悉类的声明和定义；

2.了解构造函数及其意义，并学会使用；

3.了解构造函数及其意义，并学会使用；

4.理解**this**语义。

二、实验题目

题目一 椭圆类

1.要点分析

定义椭圆类，构造函数。

2.程序源码

#include<iostream>

using namespace std;

class ellipse

{

private:

double a, b, x, y;

public:

ellipse(double aa, double bb, double xx, double yy) :a(aa), b(bb), x(xx), y(yy) {}

double area() { return 3.1415926\*a\*b; }

};

int main()

{

cout << "请输入圆心坐标：";

double a, b, x, y;

cin >> x >> y;

cout << "请输入半长轴，半短轴：";

cin >> a >> b;

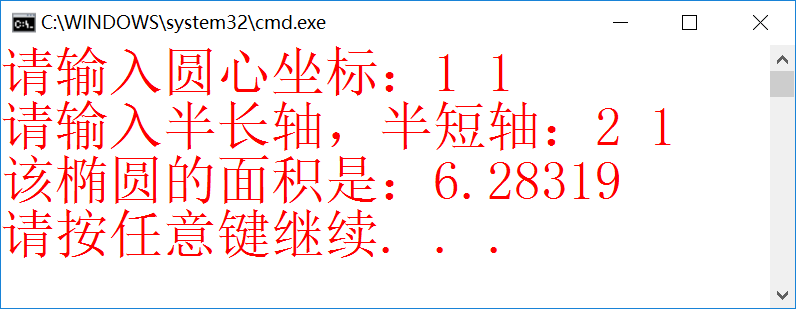
ellipse el(a, b, x, y);

cout << "该椭圆的面积是：" << el.area() << endl;

return 0;

}

3.实验结果



题目二 TIME类

1.要点分析

2.程序源码

#include<iostream>

using namespace std;

class Time

{

int hour;

int min;

int sec;

public:

Time() { hour = 0; min = 0; sec = 0; cout << "初始化：" << hour << "时" << min << "分" << sec << "秒" << endl; }

Time(int \*h, int \*m, int \*s);

Time(int \*h1, int \*m1, int \*s1, int \*h2, int \*m2, int \*s2);

void hms(int hour, int min, int sec);

void msh(int hour, int min, int sec);

};

Time::Time(int \*h, int \*m, int \*s)

{

while ((\*h)>23 || (\*h)<0 || (\*m)>59 || (\*m)<0 || (\*s)<0 || (\*s)>59)

{

cout << "输入的时间不符合格式，请重新输入";

cin >> \*h >> \*m >> \*s;

}

hour = \*h; min = \*m; sec = \*s;

cout << hour << "时" << min << "分" << sec << "秒" << endl;

}

Time::Time(int \*h1, int \*h2, int \*m1, int \*m2, int \*s1, int \*s2)

{

while ((\*h1)>23 || (\*h1)<0 || (\*m1)>59 || (\*m1)<0 || (\*s1)<0 || (\*s1)>59)

{

cout << "输入的时间不符合格式，请重新输入";

cin >> \*h1 >> \*m1 >> \*s1;

}

while ((\*h2)>23 || (\*h2)<0 || (\*m2)>59 || (\*m2)<0 || (\*s2)<0 || (\*s2)>59)

{

cout << "输入的时间不符合格式，请重新输入";

cin >> \*h2 >> \*m2 >> \*s2;

}

hour = \*h1 - \*h2; min = \*m1 - \*m2; sec = \*s1 - \*s2;

if (sec<0)

{

sec = 60 + sec;

min--;

}

if (min<0)

{

min = min + 60;

hour--;

}

cout << "时间差为:" << hour << "时" << min << "分" << sec << "秒" << endl;

}

void Time::hms(int hour, int min, int sec)

{

cout << "时间是" << hour << "时" << min << "分" << sec << "秒" << endl;

}

void Time::msh(int hour, int min, int sec)

{

cout << "时间是" << min << "分" << sec << "秒" << hour << "时" << endl;

}

int main()

{

cout << "请输入时间:";

int h1, m1, s1;

cin >> h1 >> m1 >> s1;

Time time1;

Time time2(&h1, &m1, &s1);

time2.hms(h1, m1, s1);

time2.msh(h1, m1, s1);

cout << "请输入第二个时间:";

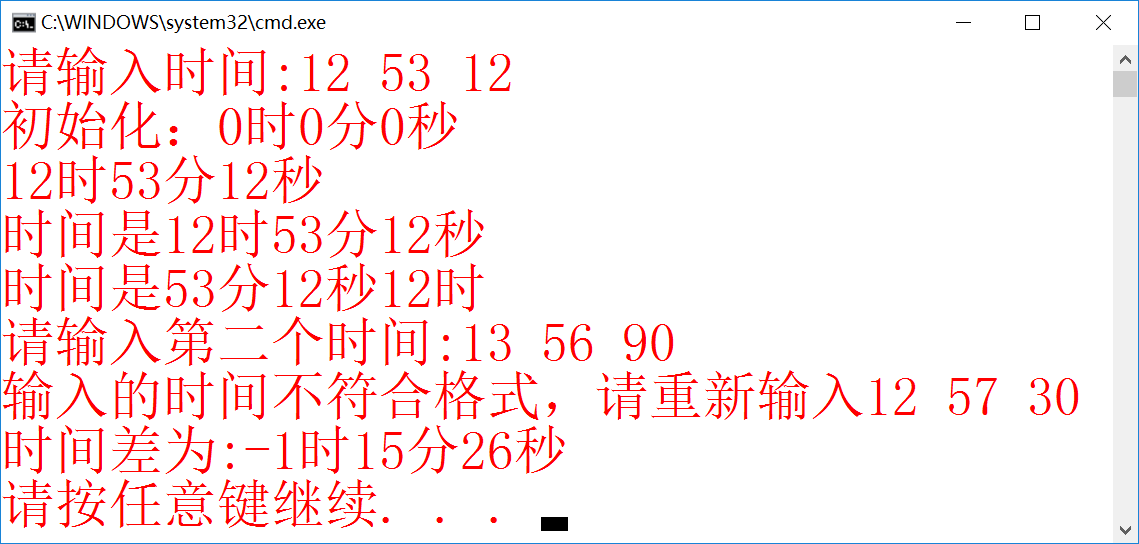
int h2, m2, s2; cin >> h2 >> m2 >> s2;

Time time6(&h1, &m1, &s1, &h2, &m2, &s2);

return 0;

}

3.实验结果



题目三 分数运算

1.要点分析

**辗转相除法；**分数化简；加减乘除运算公式

2.程序源码

#include<iostream>

using namespace std;

class rational

{

private:

int a, b;

int Gcd(int M, int N)

{

int Rem;

while (N > 0)

{

Rem = M % N;

M = N;

N = Rem;

}

return M;

}

void simp(int \*p, int \*q)

{

int a;

a = Gcd(\*p, \*q);

\*p = (\*p) / a;

\*q = (\*q) / a;

}

public:

rational() :a(0), b(1) {}

void add(int a1, int b1, int a2, int b2);

void mult(int a1, int b1, int a2, int b2);

void sub(int a1, int b1, int a2, int b2);

void div(int a1, int b1, int a2, int b2);

void printfrac()

{

if (b != 1)

{

cout << a << "/" << b << endl;

}

else cout << a;

}

void printflo() { cout << a / b << endl; }

};

void rational::add(int a1, int b1, int a2, int b2)

{

simp(&a1, &b1);

simp(&a2, &b2);

a = a1\*b2 + a2\*b1;

b = b1\*b2;

simp(&a, &b);

}

void rational::mult(int a1, int b1, int a2, int b2)

{

simp(&a1, &b1);

simp(&a2, &b2);

simp(&a1, &b2);

simp(&a2, &b1);

a = a1\*a2;

b = b1\*b2;

}

void rational::sub(int a1, int b1, int a2, int b2)

{

simp(&a1, &b1);

simp(&a2, &b2);

a = a1\*b2 - a2\*b1;

b = b1\*b2;

simp(&a, &b);

}

void rational::div(int a1, int b1, int a2, int b2)

{

simp(&a1, &b1);

simp(&a2, &b2);

simp(&a1, &b2);

simp(&a2, &b1);

a = a1\*b2;

b = b1\*a2;

}

int main()

{

cout << "请输入运算类型：";

char str[10];

cin >> str;

while (strcmp(str, "加法") != 0 && strcmp(str, "减法") != 0 && strcmp(str, "乘法") != 0 && strcmp(str, "除法") != 0)

{

cout << "您的输入不符合要求，请重新输入：";

cin >> str;

}

int a1, a2, b1, b2;

cout << "请输入第一个分数的分子和分母：";

cin >> a1 >> b1;

cout << "请输入第二个分数的分子和分母：";

cin >> a2 >> b2;

char way[10];

cout << "请输入输出方式：";

cin >> way;

while (strcmp(way, "分数") != 0 && strcmp(way, "浮点数") != 0)

{

cout << "您的输入不符合要求，请重新输入：";

cin >> way;

}

rational rat;

if (strcmp(str, "加法") == 0)

{

cout << a1 << "/" << b1 << "+" << a2 << "/" << b2 << "=";

rat.add(a1, b1, a2, b2);

}

if (strcmp(str, "减法") == 0)

{

cout << a1 << "/" << b1 << "-" << a2 << "/" << b2 << "=";

rat.sub(a1, b1, a2, b2);

}

if (strcmp(str, "乘法") == 0)

{

cout << a1 << "/" << b1 << "\*" << a2 << "/" << b2 << "=";

rat.mult(a1, b1, a2, b2);

}

if (strcmp(str, "除法") == 0)

{

cout << "(" << a1 << "/" << b1 << ")" << "/" << "(" << a2 << "/" << b2 << ")" << "=";

rat.div(a1, b1, a2, b2);

}

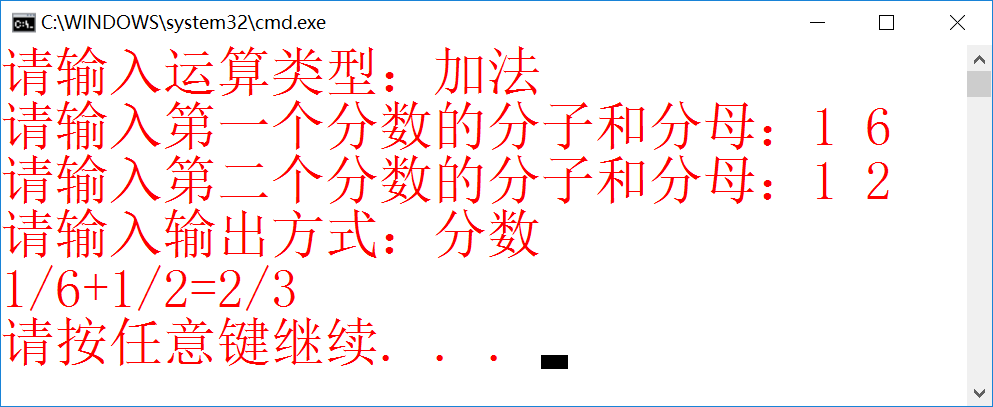
if (strcmp(way, "分数") == 0)rat.printfrac();

else rat.printflo();

return 0;

}

3.实验结果



题目四 小车类

1.要点分析

略

2.程序源码

#include<iostream>

#include<windows.h>

#include <mmsystem.h>

#pragma comment(lib,"winmm.lib")

using namespace std;

char \*songs(int n)

{

static char\*songs[] =

{ "第一首歌","第二首歌","第三首歌","第四首歌","第五首歌","第六首歌","第七首歌","第八首歌","第九首歌","第十首歌" };

return songs[n];

}

class automobile

{

char shape[20];

char type[20];

char color[20];

float price;

int carry\_weight;

int carry\_customer;

char \*qm[10];

public:

automobile(char \*s, char \*t, char \*c, float pri, int cw, int cc);

void movecar(int l, int k);

void horming(int num);

void downcar();

void backcar();

void slopecar();

void turnright();

void choose();

void play\_mp3(char \*ps);

char \*show\_type() { return type; }

~automobile();

};

automobile::automobile(char \*s, char \*t, char \*c, float pri, int cw, int cc)

{

strcpy(shape, s);

strcpy(type, t);

strcpy(color, c);

price = pri;

carry\_weight = cw;

carry\_customer = cc;

for (int i = 0; i<10; i++)

qm[i] = songs(i);

}

void automobile::movecar(int l, int k)

{

cout << "\n" << type << "水平直线运动:" << endl;

for (int i = 0; i<l; i++)

{

int s = strlen(shape);

cout << ' ' << shape;

Sleep(1000 / k);

for (int j = 0; j<s; j++)

cout << '\b';

}

}

void automobile::downcar()

{

cout << "\n" << type << "垂直下降运动:" << endl;

for (int i = 0; i<10; i++)

{

cout << shape;

Sleep(500);

cout << '\r';

int s = strlen(shape);

for (int j = 0; j<s; j++)

cout << ' ';

cout << endl;

}

}

void automobile::backcar()

{

cout << "\n" << type << "倒车运动:" << endl;

for (int i = 0; i<15; i++)

cout << ' ';

for (int i = 0; i<10; i++)

{

cout << shape;

Sleep(1000);

int s = strlen(shape);

for (int j = 0; j<s; j++)

cout << '\b';

for (int j = 0; j<s; j++)

cout << ' ';

for (int j = 0; j<s; j++)

cout << '\b';

cout << '\b';

}

cout << endl;

}

void automobile::slopecar()

{

cout << "\n" << type << "斜向下运动:" << endl;

int k = 2;

for (int i = 0; i<10; i++)

{

cout << shape;

Sleep(1000);

int s = strlen(shape);

for (int j = 0; j<s; j++)

cout << '\b';

for (int j = 0; j<s; j++)

cout << ' ';

cout << endl;

for (int j = 0; j<k; j++)

cout << ' ';

k += 2;

}

cout << endl;

}

void automobile::turnright()

{

cout << "\n" << type << "右拐运动:" << endl;

for (int k = 0; k<5; k++)

{

for (int i = 0; i<10; i++)

{

int s = strlen(shape);

cout << ' ' << shape;

Sleep(500);

for (int j = 0; j<s; j++)

cout << '\b';

}

for (int i = 0; i<10; i++)

{

cout << shape;

Sleep(500);

cout << '\r';

int s = strlen(shape);

for (int j = 0; j<s + 10 \* (k + 1); j++)

cout << ' ';

cout << endl;

for (int j = 0; j<10 \* (k + 1); j++)

cout << ' ';

}

}

}

void automobile::horming(int num)

{

for (int i = 0; i<num; i++)

{

cout << type;

cout << '\007' << " di..." << endl;

Sleep(1000);

}

}

void automobile::play\_mp3(char \*ps)

{

char str[100] = "play ";

strcat(str, ps);

cout << str;

}

void automobile::choose()

{

cout << "\n所有歌曲如下：\n";

cout << "曲目 歌名\n";

cout << "-------------------\n";

for (int i = 0; i<10; i++)

cout << " " << i + 1 << " " << qm[i] << endl;

int n;

cout << "\n请输入数字~10选择需要播放的歌曲。\n现在请输入数字：";

cin >> n;

while (n <= 0 || n>10)

{

cout << "输入数字不在范围内，请重新输入：";

cin >> n;

}

play\_mp3(qm[n - 1]);

}

automobile::~automobile()

{

cout << "\n已撤销以上变量。\n";

}

int main()

{

char tp[] = "yyida";

char ys[] = "white";

char xz[20];

cout << "请输入小车形状：";

cin >> xz;

automobile tesila(xz, tp, ys, 200000, 5, 3);

tesila.horming(5);

tesila.movecar(10, 3);

tesila.downcar();

tesila.backcar();

tesila.slopecar();

tesila.turnright();

tesila.choose();

return 0;

}

题目五 圆管类

1.要点分析

基本没有

2.程序源码

#include<iostream>

using namespace std;

const float pi = 3.1415926;

class pipe

{

private:

float r, R, h, den;

public:

pipe(float rr, float RR, float hh, float dd) :r(rr), R(RR), h(hh), den(dd) {}

float volume() { return pi\*R\*R\*h; }

float cubage() { return pi\*r\*r\*h; }

float sarea() { return 2 \* pi\*(R\*R - r\*r) + 2 \* pi\*h\*(r + R); }

float weight() { return den\*pi\*h\*(R\*R - r\*r); }

void print()

{

cout << "容积：" << cubage() << endl;

cout << "总体积：" << volume() << endl;

cout << "表面积：" << sarea() << endl;

cout << "重量：" << weight() << endl;

}

~pipe() { cout << "堆区空间已释放" << endl; }

};

int main()

{

float r, R, h, den;

cout << "请输入水管内径：";

cin >> r;

cout << "请输入水管外径：";

cin >> R;

while (r > R)

{

cout << "数据输入有误，请重新输入" << endl;

cout << "请输入水管内径：";

cin >> r;

cout << "请输入水管外径：";

cin >> R;

}

cout << "请输入水管长度：";

cin >> h;

cout << "请输入水管材料密度：";

cin >> den;

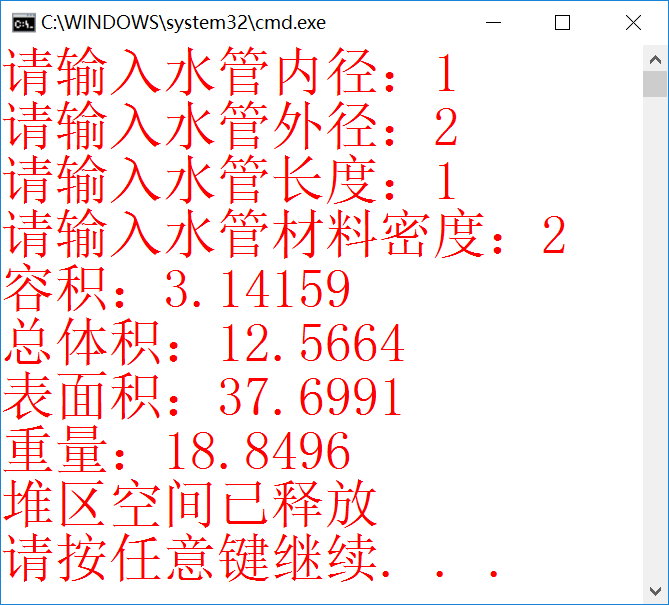
pipe p(r, R, h, den);

p.print();

return 0;

}

3.实验结果



题目六 方程组类

1.要点分析

在上周类中加入相加方法。

2.程序源码

#include<iostream>

#include<cmath>

using namespace std;

class equationset

{

private:

float a, b, c, d, e, f, x, y;

public:

equationset(float aa, float bb, float cc, float dd, float ee, float ff)

{

set(aa, bb, cc, dd, ee, ff);

}

void set(float aa, float bb, float cc, float dd, float ee, float ff)

{

a = aa; b = bb; c = cc; d = dd; e = ee; f = ff;

}

void calculate();

void showset();

void showans();

void add(equationset e1)

{

a = a + e1.a;

b = b + e1.b;

c = c + e1.c;

d = d + e1.d;

f = f + e1.f;

e = e + e1.e;

}

void dda(equationset e1)

{

a = a - e1.a;

b = b - e1.b;

c = c - e1.c;

d = d - e1.d;

f = f - e1.f;

e = e - e1.e;

}

};

void equationset::calculate()

{

if (a\*e == b\*d)

{

if (a / d == c / f)

{

x = a\*b\*c \* 12345; //防止巧合

}

else

{

x = f\*d\*e \* 54321;

}

}

else

{

x = (c\*e - b\*f) / (a\*e - b\*d);

y = (a\*f - c\*d) / (a\*e - b\*d);

}

}

void equationset::showset()

{

cout << "方程组为:\n";

if (a != 0)

{

cout << a << "x";

}

if (b>0)

{

if (a != 0)

{

cout << "+" << b << "y=";

}

else

{

cout << b << "y=";

}

}

else

{

if (b != 0)

{

cout << b << "y=";

}

else

{

cout << "=";

}

}

cout << c << endl;

if (d != 0)

{

cout << d << "x";

}

if (e>0)

{

if (d != 0)

{

cout << "+" << e << "y=";

}

else

{

cout << e << "y=";

}

}

else

{

if (e != 0)

{

cout << e << "y=";

}

else

{

cout << "=";

}

}

cout << f << endl;

}

void equationset::showans()

{

if (x == a\*b\*c \* 12345)

{

cout << "有无数解。\n";

}

else

{

if (x == d\*e\*f \* 54321)

{

cout << "无解。\n";

}

else

{

cout << "解为:\nx=" << x << "\ny=" << y << endl;

}

}

}

int main()

{

float a, b, c, d, e, f;

cout << "请依次输入第一个二元一次方程组六个系数,分别按顺序对应\nAX + BY = C\nDX + EY = F\n中的a,b,c,d,e,f:\n";

cin >> a >> b >> c >> d >> e >> f;

equationset e1(a, b, c, d, e, f);

e1.showset();

e1.calculate();

e1.showans();

cout << "请依次输入第二个二元一次方程组六个系数,分别按顺序对应\nAX + BY = C\nDX + EY = F\n中的a,b,c,d,e,f:\n";

cin >> a >> b >> c >> d >> e >> f;

equationset e2(a, b, c, d, e, f);

e2.showset();

e2.calculate();

e2.showans();

e1.add(e2);

cout << "两个方程组相加后为:\n";

e1.showset();

e1.calculate();

e1.showans();

e2.dda(e1);

cout << "第二个方程组减去第一个后为:\n";

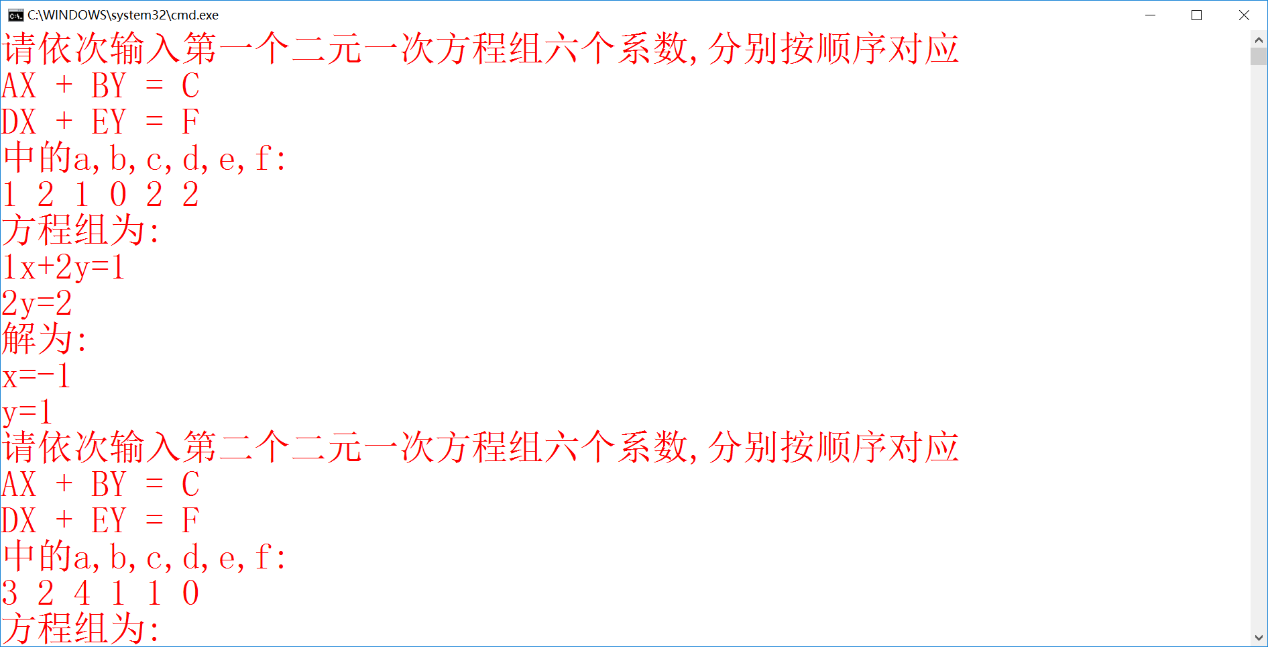
e2.showset();

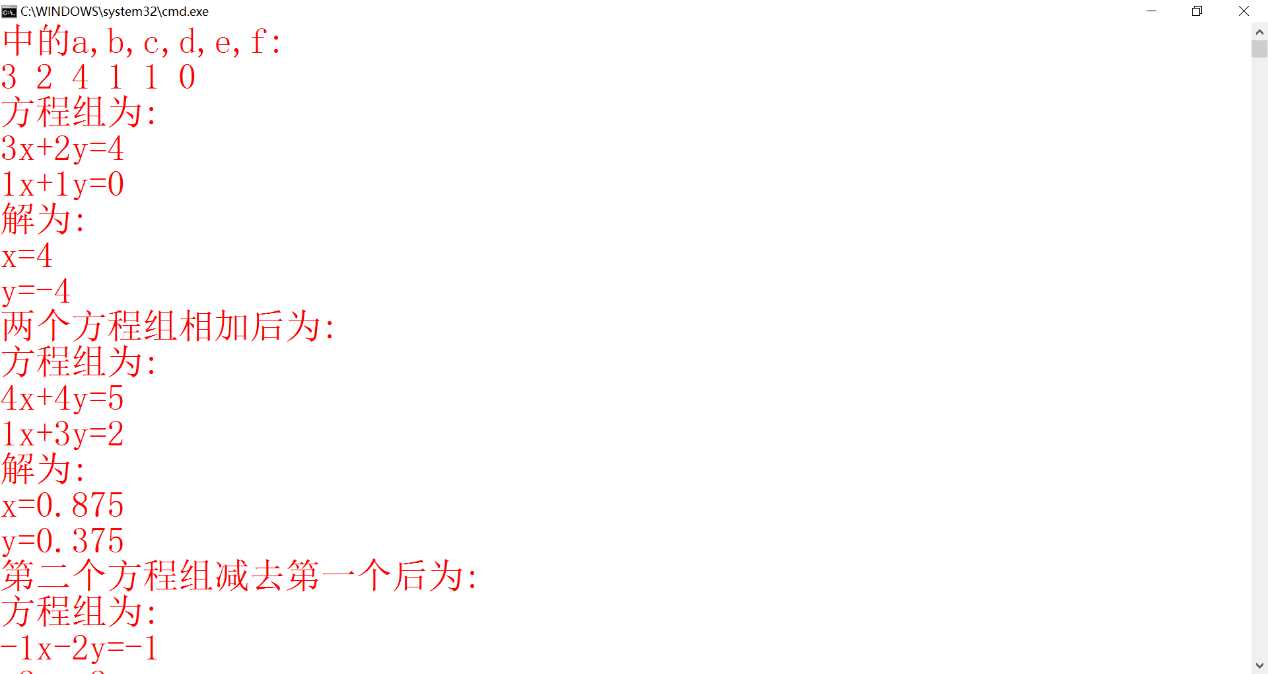
e2.calculate();

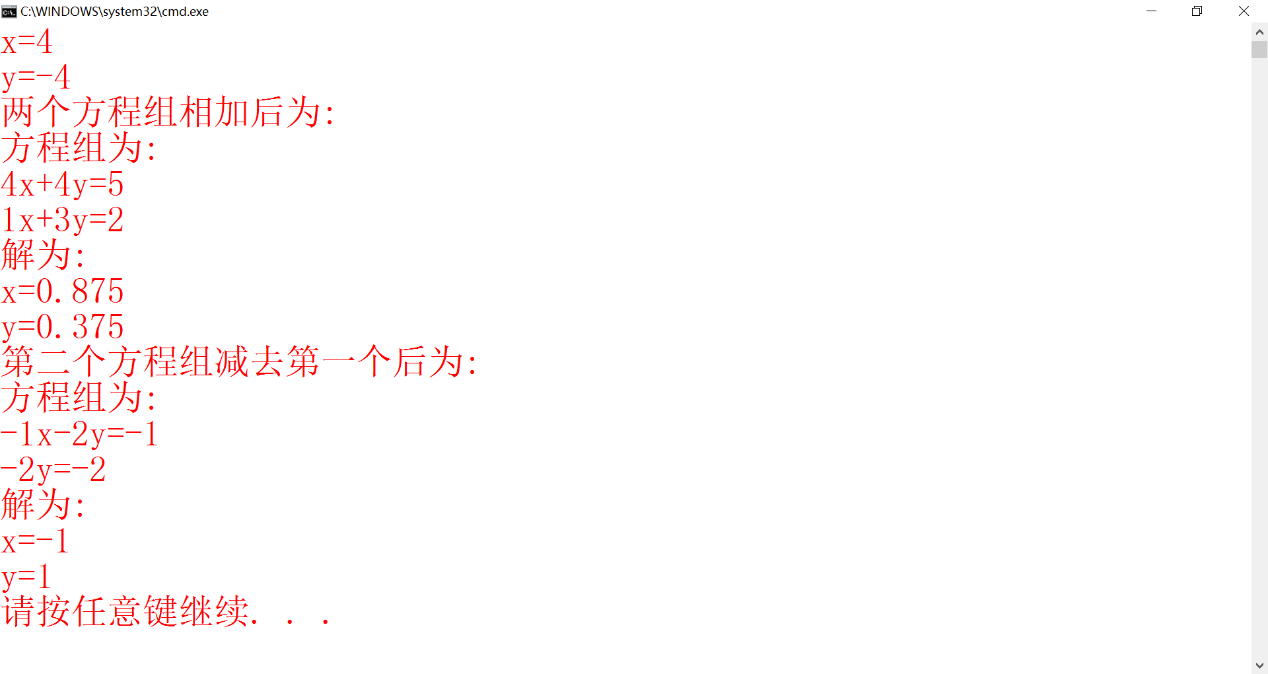
e2.showans();

}

3.实验结果







题目七 机器人

1.要点分析

将前几次的作业放在一个类里面

2.程序源码

#include<iostream>

#include<cstring>

#include<cmath>

using namespace std;

class Robot

{

char name[20];

int age;

int num;

char hanzi[10];

void trge(int a)//将一位数转化为相应的英文

{

if (a == 1) cout << "one" << ' ';

else if (a == 2) cout << "two" << ' ';

else if (a == 3) cout << "three" << ' ';

else if (a == 4) cout << "four" << ' ';

else if (a == 5) cout << "five" << ' ';

else if (a == 6) cout << "six" << ' ';

else if (a == 7) cout << "seven" << ' ';

else if (a == 8) cout << "eight" << ' ';

else if (a == 9) cout << "nine" << ' ';

}

void trshi(int x)//将2位数转化为相应的英文（几十几）

{

int a = x / 10, b = x % 10;

if (a == 0)

{

if (b != 0) trge(b);//若为个位数则按上一个函数输出

}

else if (a == 1)//十位数为1的情况

{

if (b == 0) cout << "ten" << ' ';

if (b == 1) cout << "eleven" << ' ';

if (b == 2) cout << "twelve" << ' ';

if (b == 3) cout << "thirteen" << ' ';

if (b == 4) cout << "fourteen" << ' ';

if (b == 5) cout << "fifteen" << ' ';

if (b == 6) cout << "sixteen" << ' ';

if (b == 7) cout << "seventeen" << ' ';

if (b == 8) cout << "eighteen" << ' ';

if (b == 9) cout << "nighteen" << ' ';

}

else if (a == 2)

{

cout << "twenty" << ' '; trge(b);

}

else if (a == 3)

{

cout << "thirty" << ' '; trge(b);

}

else if (a == 4)

{

cout << "forty" << ' '; trge(b);

}

else if (a == 5)

{

cout << "fifty" << ' '; trge(b);

}

else if (a == 6)

{

cout << "sixty" << ' '; trge(b);

}

else if (a == 7)

{

cout << "seventy" << ' '; trge(b);

}

else if (a == 8)

{

cout << "eighty" << ' '; trge(b);

}

else if (a == 9)

{

cout << "ninety" << ' '; trge(b);

}

}

void trbai(int y)//将三位数转化为英文

{

int a, b, c;

a = y / 100;

b = y - 100 \* a;//分离百位和后两位

if (a)//若a不为0

{

trge(a);

cout << "hundred" << ' '; trshi(b);

}

else trshi(b);

}

public:

Robot(char list[], int a, int b)

{

strcpy(name, list);

age = a;

num = b;

char hanzi1[] = "曲王平大作";

strcpy(hanzi, hanzi1);

}

~Robot() { cout << "\n已成功析构"<<endl; }

void trans()

{

cout << "整数翻译结果：";

int a, b, c, d;

int x = num;

a = x / 1000000000; b = (x - a \* 1000000000) / 1000000; d = x % 1000; c = (x / 1000) % 1000;//按分节符分离各三位数

if (a != 0) { trbai(a); cout << "billion" << ' '; }

if (b != 0) { trbai(b); cout << "million" << ' '; }

if (c != 0) { trbai(c); cout << "thousand" << ' '; }

trbai(d);//输出结果

cout << endl;

}

void writechn(int i)

{

switch (i)

{

case 1:

{

cout << " \* \* " << endl;

cout << " \* \* " << endl;

cout << "\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "\* \* \* \*" << endl;

cout << "\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "\* \* \* \*" << endl;

cout << "\*\*\*\*\*\*\*\*\*\*" << endl;

break;

}

case 2:

{

cout << "\*\*\*\*\*\*\*\*\*\*" << endl;

cout << " \*\* " << endl;

cout << "\*\*\*\*\*\*\*\*\*\*" << endl;

cout << " \*\* " << endl;

cout << "\*\*\*\*\*\*\*\*\*\*" << endl;

break;

}

case 3:

{

cout << " \*\* " << endl;

cout << "\*\*\*\*\*\*\*\*\*\*" << endl;

cout << " \*\* " << endl;

cout << "\*\*\*\*\*\*\*\*\*\*" << endl;

cout << " \*\* " << endl;

cout << "\*\*\*\*\*\*\*\*\*\*" << endl;

cout << " \*\* " << endl;

break;

}

case 4:

{

cout << " \* " << endl;

cout << " \* " << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << " \* \* " << endl;

cout << " \* \* " << endl;

cout << " \* \* " << endl;

cout << " \* \* " << endl;

break;

}

case 5:

{

cout << "\*\*\*\*\*\*\*\*\*\*" << endl;

cout << " \* \* \* " << endl;

cout << " \* \* \* " << endl;

cout << "\*\*\*\*\*\*\*\*\*\*" << endl;

cout << " \* " << endl;

cout << " \* " << endl;

break;

}

}

}

void jdg()

{

int n = num;

if (n % 2 == 0)

cout << "该整数是偶数" << endl;

else

cout << "该整数是奇数" << endl;

int count = 0; int tmp = sqrt(num);

for (int i = 2; i<tmp + 1; i++)

{

if (n%i == 0)

{

count++;

break;

}

}

if (count == 0)

cout << "该整数是素数";

else

cout << "该整数不是素数" << endl;

int list[9];

int k = 0;

while (n != 0)

{

list[k] = n % 10; n = n / 10; k++;

}

int count2 = 0;

for (int j = 0; j<k - 1; j++)

{

if (list[j]<list[j + 1])

{

count2++; break;

}

}

if (count2 == 0)

cout << "该整数是递增数:" << endl;

else

cout << "该整数不是递增数" << endl;

int count3 = 0;

for (int m = 0; m<k / 2; m++)

{

if (list[m] != list[k - m - 1])

{

count3++;

break;

}

}

if (count3 == 0)

cout << "该数字是回文数";

else

cout << "该数字不是回文数" << endl;

}

};

int main()

{

cout << "请输入机器人的姓名、年龄和整数:"<<endl;

char nm[11];

int age1, num1;

cin.get(nm, 10);

cin >> age1 >> num1;

Robot bot1(nm, age1, num1);

cout << "整数翻译结果:"; bot1.trans();

cout << "\n请输入想要输出的汉字的编号(支持曲、王、丰、大、平五个字);";

int n;

cin >> n;

cout << endl;

bot1.writechn(n);

cout << endl << "整数性质判断结果是:" << endl;

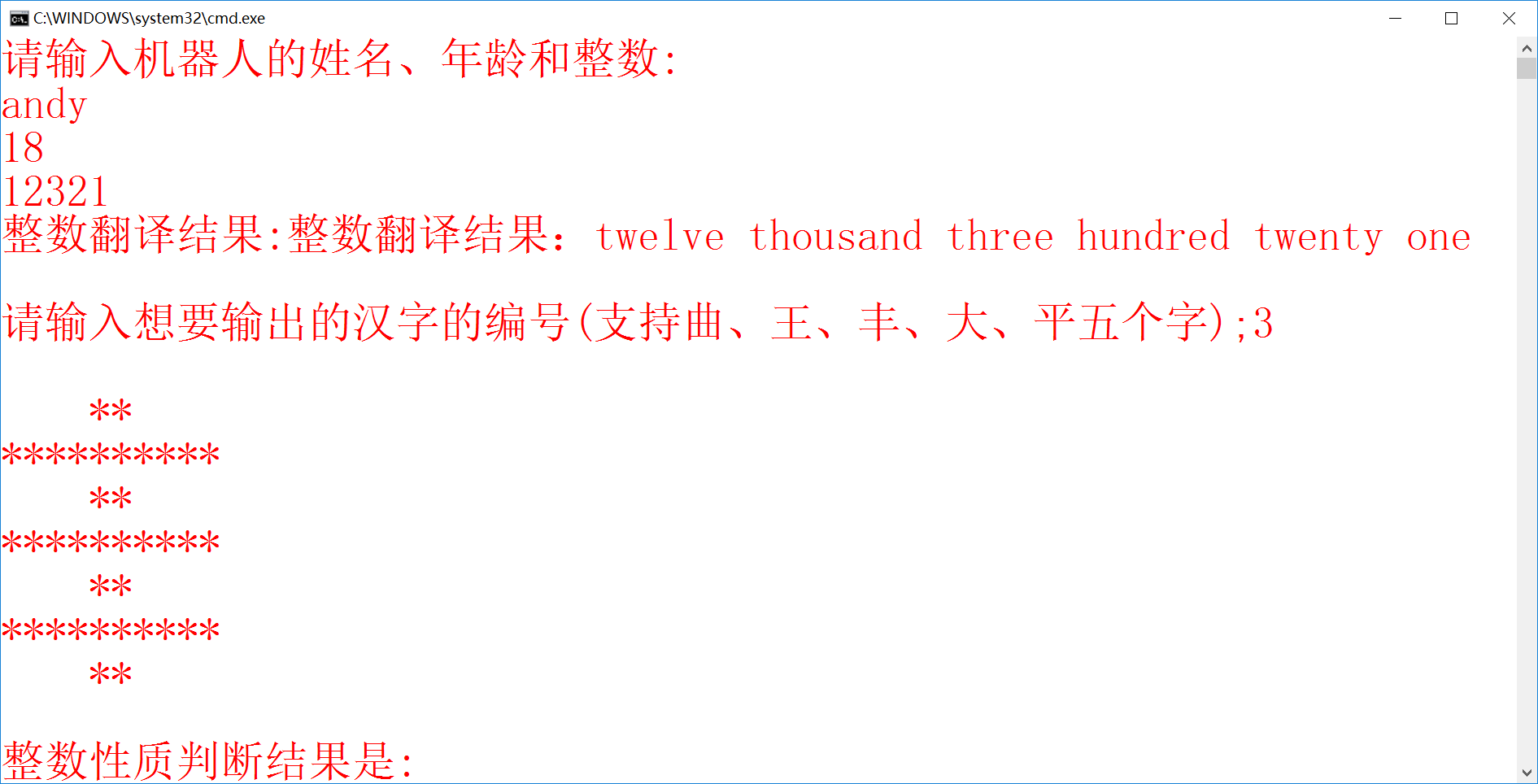
bot1.jdg();

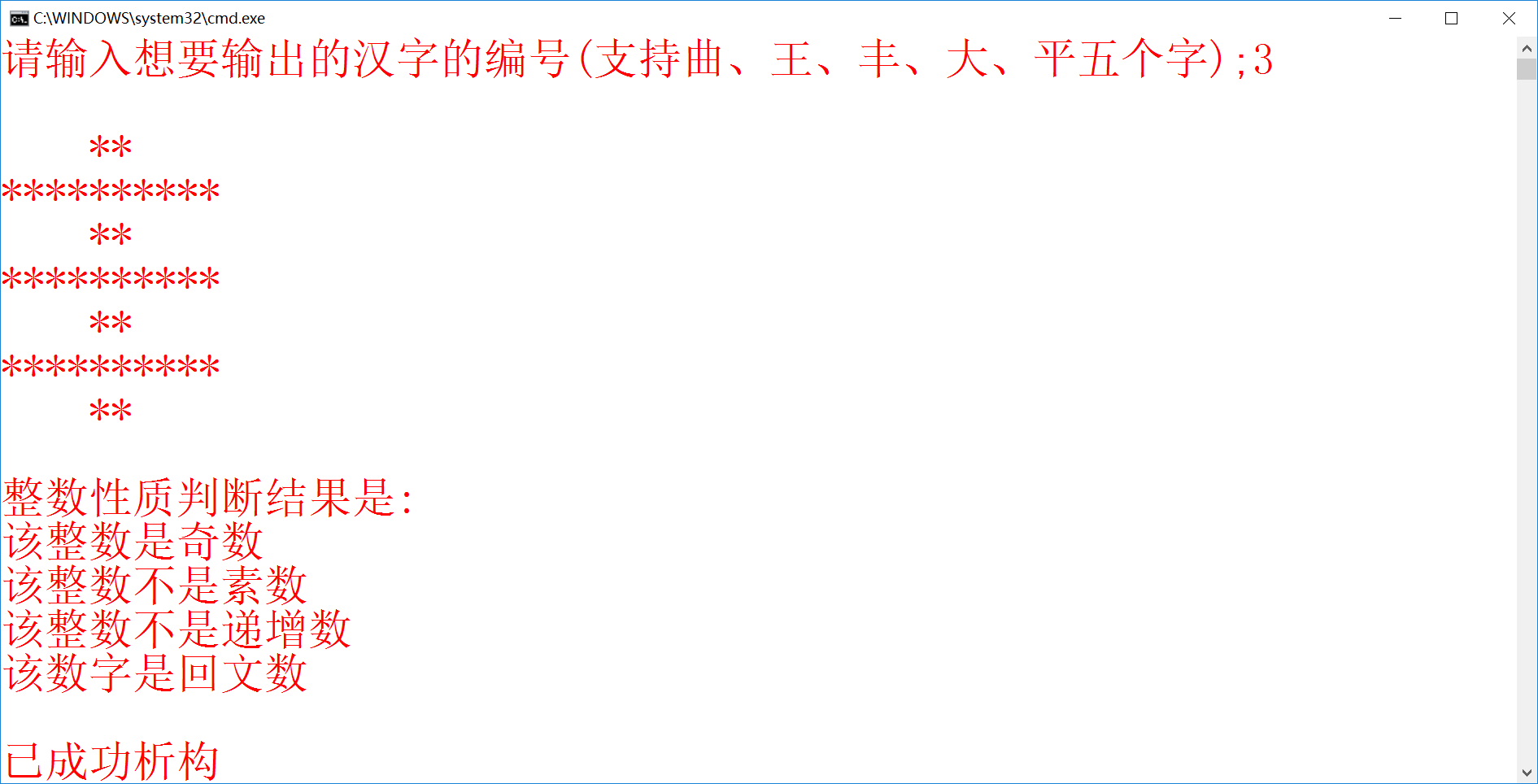
cout << endl;

return 0;

}

3.实验结果





三、分析对比

本周又拜读了学长郭昱博同志的实验报告。何谓又呢？显然是因为不是第一次了。然而即使看了再多次，震撼仍是震撼，仰望的角度一直不变。从浅了讲，看的是排版——字号、颜色到一级标题、二级标题，标准的美工底子，天生的设计师；然而更深点看，则更加耐人寻味：首先，学长每次标明，本次实验花了六小时——一方面是给老师的反馈，另一方面也逼的我心里算一笔账：自己每次花多久写实验？抖抖索索把零碎时间加上，五个小时是跑不掉的：这样一算脸上就难看了起来，工作量只有别人的一半，花的时间却没少，如此只得长喟叹，自己离熟练还差把火候。另外，学长的实验报告里里外外还透露着一种气息：细。这一点从输入提示文字，数据合理性的检测及反馈，程序注释里无一不在体现。最后，学长常有一题多解，这对我简直是天方夜谭——无论黑猫白猫，做出题目就是好猫，哪还轮得到分析优劣？早就丢到一边出去玩了。

期中的阴影还没过去，新的风暴又要出现。人生加勉之处多矣，有人说大学生活就是“间接性踌躇满志，持续性混吃等死”。我不同意，我也不会同意。