类与对象

一、#include<iostream>

using namespace std;

class Time

{

private:

int hour;

int minute;

int sec;

public:

void SetTime(void)

{

cin >> hour >> minute >> sec;

}

void ShowTime(void)

{

cout << hour <<":" << minute<<":" << sec;

}

};

int main()

{

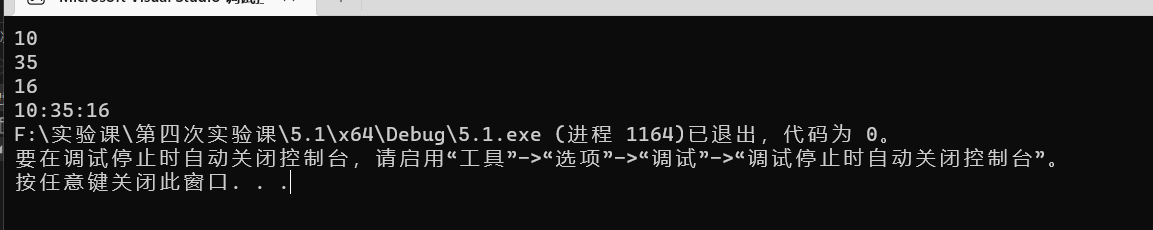
Time t1;

t1.SetTime();

t1.ShowTime();

return 0;

}

运行结果

二、#include<iostream>

#include"student.h"

using namespace std;

void Student::display()

{

cout << "num：" << num << endl;

cout << "name：" << name << endl;

cout << "sex：" << sex << endl;

}

void Student::set\_value(void)

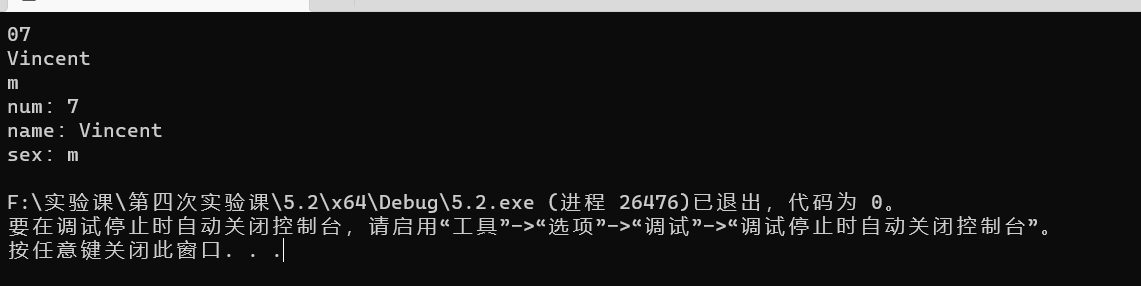
{

cin >> num;

cin >> name;

cin >> sex;

}

运行结果

三、#include<iostream>

using namespace std;

class Column

{

private:

float a;

float b;

float c;

float v;

public:

void set\_value();

float j();//计算长方柱的体积

void show\_value();

};

void Column::set\_value()

{

cin >> a;

cin >> b;

cin >> c;

}

float Column::j()

{

v= a \* b \* c;

return v;

}

void Column::show\_value()

{

cout << v << endl;

}

int main()

{

Column c1, c2, c3;

cout << "请输入第一个长方柱的长，宽，高" << endl;

c1.set\_value();

c1.j();

cout << "第一个长方柱体积为：";

c1.show\_value();

cout << "请输入第二个长方柱的长，宽，高" << endl;

c2.set\_value();

c2.j();

cout << "第二个长方柱体积为：";

c2.show\_value();

cout << "请输入第三个长方柱的长，宽，高" << endl;

c3.set\_value();

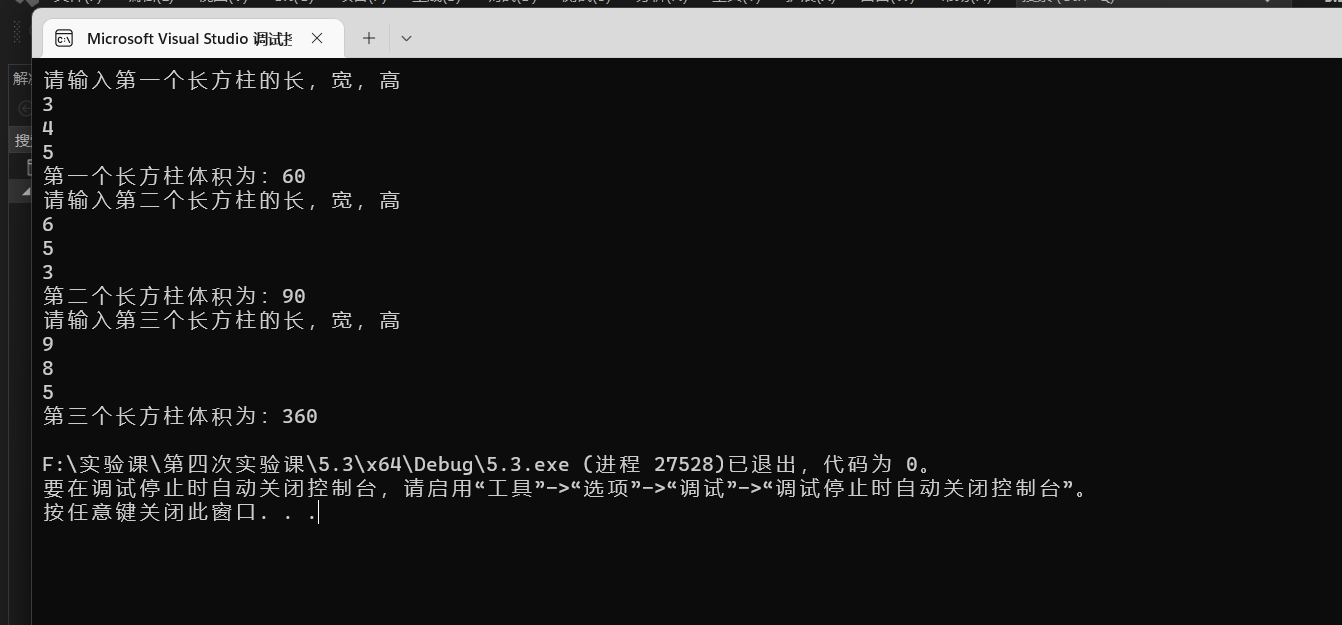
c3.j();

cout << "第三个长方柱体积为：";

c3.show\_value();

return 0;

}

运行结果

四、

#include<iostream>

#include<math.h>

using namespace std;

class Shape

{

public:

virtual double S()

{

return 0;

};

};

class Circle :public Shape

{

public:

const double PI = 3.14;

double r;

Circle(double r)

{

this->r = r;

}

virtual double S()

{

return PI\*r\*r;

}

};

class Square:public Shape

{public:

double side;

Square(double side)

{

this->side = side;

}

double S()

{

return side \* side;

}

};

class Rectangle :public Shape

{public:

double length, width;

Rectangle(double length, double width)

{

this->length = length;

this->width = width;

}

double S()

{

return length \* width;

}

};

class Trapezoid :public Shape

{public:

double sd, xd, h;//梯形的上底下底和高

Trapezoid(double sd, double xd, double h)

{

this->sd = sd;

this->xd = xd;

this->h = h;

}

double S()

{

return (sd + xd) \* h \* 0.5;

}

};

class Triangle :public Shape

{public:

double l1, l2, l3;

Triangle(double l1,double l2,double l3)

{

this->l1 = l1;

this->l2 = l2;

this->l3 = l3;

}

double S()

{

double P;

P = (l1 + l2 + l3) \* 0.5;

return sqrt(P\*(P - l1)\*(P - l2)\*(P - l3));

}

};

int main()

{

Circle circle(10.2);

Square square(3);

Rectangle rectangle(3, 4);

Trapezoid trapezoid(2.0, 4.5, 3);

Triangle triangle(4, 5, 6);

Shape\* p[5] = { &circle,&square,&rectangle,&trapezoid,&triangle };

cout << "圆形面积为" << p[0]->S() << endl;

cout << "正方形面积为" << p[1]->S() << endl;

cout << "长方形面积为" << p[2]->S() << endl;

cout << "梯形面积为" << p[3]->S() << endl;

cout << "三角形面积为" << p[4]->S() << endl;

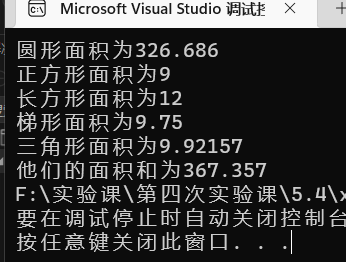
double H;

H = (p[0]->S()) + (p[1]->S()) + (p[2]->S()) + (p[3]->S()) + (p[4]->S());

cout << "他们的面积和为" << H;

return 0;

}

运行结果

五、没有思路，查找资料也没找到，只能作罢。

遇到的问题与解决方法：

写类的继承时，很多知识点都忘记了，只能把课本又重新看了一遍，慢慢又明白了该怎么样去写派生。

感悟：温故而知新。