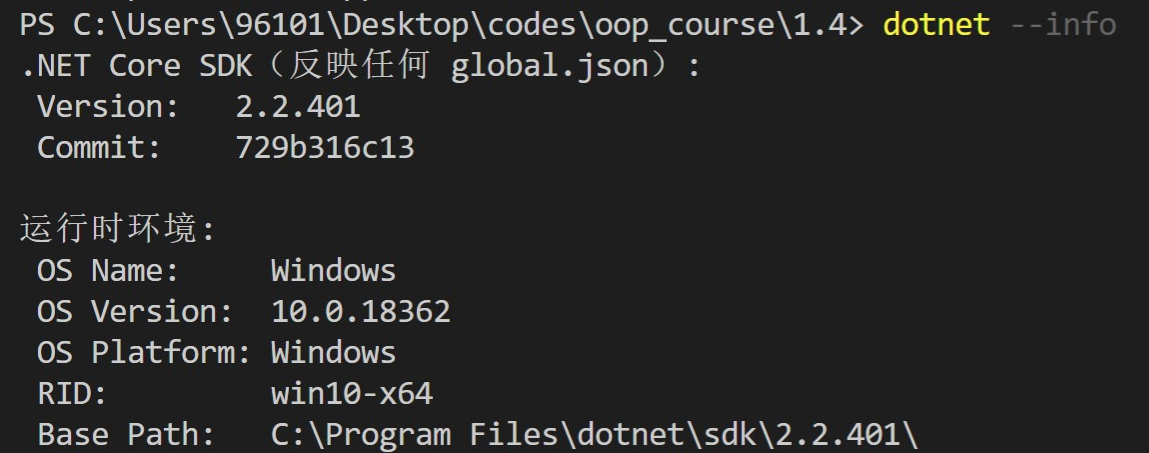
**实验一 C#程序基本结构**

实验目的：1. 熟悉VS 的IDE操作环境；掌握调试工具的使用；

2. 练习编写C#控制台程序；掌握控制台输入、输出的方法；

任务：

运行环境：



1. 上机完成习题1.1.
2. 学习对象浏览器的使用
   1. 利用对象浏览器查看Console类的成员
   2. 了解Console的成员函数Write(), WriteLine(), Read(), ReadLine(), ReadKey();

WriteLine()函数一共有 18 个重载函数?

Read()、ReadLine() 和 ReadKey()分别接收什么类型的数据？

Read()接收一个字符即字符型数据，返回该字符的Unicode码；ReadLine()接受的是一串字符串即string；ReadKey()接受的是一个自定义的结构体ConsoleKeyInfo。

1. 编写一个类 Program, 其中放置Main() 函数，
2. 在Main()函数中先写入如下语句：

int I1=10;

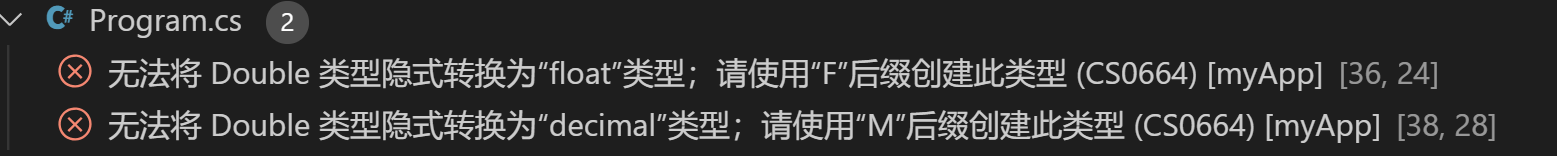
long L1=10000000000;

float F1=3.14;

double D1=6.28;

decimal Deci=9.42;

若错误列表窗提示错误，把错误代码和说明记录在下面，并按提示修正错误。



float F1 = 3.14和decimal Deci = 9.42错误，应改为float F1=3.14F和decimal Deci=9.42M。 分析：给一个浮点数变量赋值，没有加任何后缀则编译器会自动把这个数字认定为double类型，所以给float类型的变量赋值时要在数字后面加F，给decimal类型变量赋值时要在数字后面加M。

1. 尝试把long型变量赋给int型变量， float型变量赋给double型变量，再把double型变量赋给float型变量，最后把float型赋给decimal型：

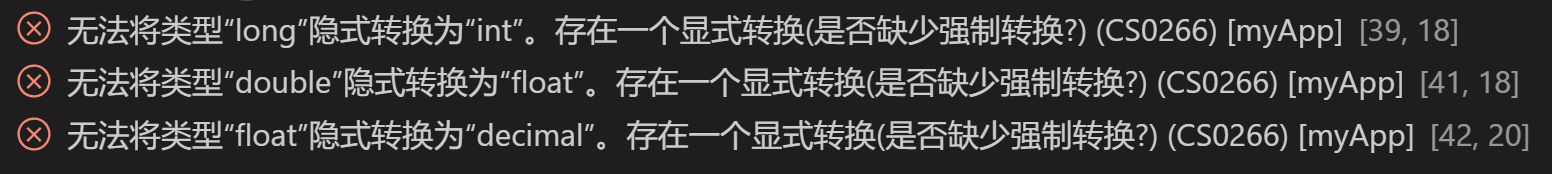
I1 = L1;

D1 = F1;

F1 = D1;

Deci = F1;

若错误列表窗提示错误，把错误代码和说明记录在下面，并按提示修正错误。



I1 = L1; F1 = D1; Deci = F1错误。应改为I1 = (int)L1; F1 = (float)D1; Deci = (decimal)F1;。

分析：当被转换类型的值范围小于目标类型的值范围时可以执行隐式转换，否则隐式转换是编译器会报异常，而且被转换类型的值必须与目标类型兼容。上面的错误中long类型的表示范围比int大，double类型的表示范围比float大，float类型和decimal类型不兼容，所以无法进行隐式转换。

1. 最后使用Write Line()函数屏幕上输出这些变量的值：

Console.WriteLine("I1={0},L1={1},F1={2},D1={3},Deci={4}",I1,L1,F1,D1,Deci );

这是WrtieLine函数的哪个重载？ 写出此重载的签名：

WriteLine ( string format, params object[] arg )

编译并运行程序，输出结果记录在下面。

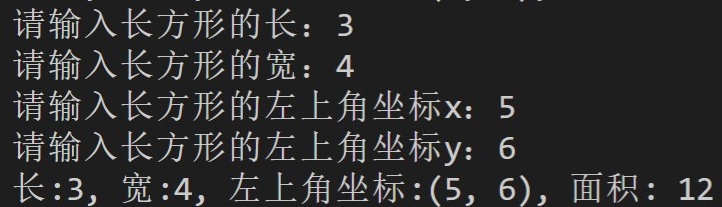
I1=1410065408,L1=10000000000,F1=3.14,D1=3.14000010490417,Deci=3.14

I1和D1的输出值合理吗？解释一下： 不合理。因为(1)int值的范围为有符号 32 位整数，最大为2147483647，而10000000000>2147483647，所以进行转换的时候会出现异常，导致I1不合理。(2)double类型的精度比float类型的精度高，小数点后需要补位，而在对小数点后的数据进行补位的时候可能会产生误差，计算机将十进制转换为二进制显示的时候是用乘二取整的方法实现的，0.14这部分无法在精度范围内通过乘二取整操作转换到1的整数倍，所以会出现误差。

修改习题1.4： 增加成员字段表示矩形面积，并增加计算矩形面积的成员方法Area(),,在原有的输出矩形信息的成员函数中增加输出矩形面积；在Main()函数中，增加从控制台接受用户输入矩形参数的代码(利用ReadLine()函数)，根据用户输入的矩形参数，输出对应的矩形信息。

修改过程顺利吗? 如果不顺利，遇到什么主要困难，找到对策了吗？

顺利



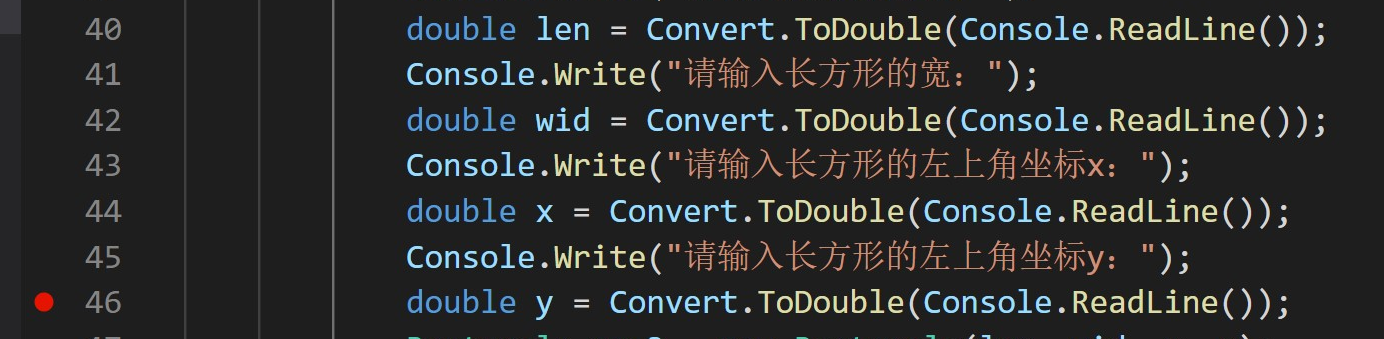
5. 调试技能学习：

a) 会利用错误列表窗定位错误

b) 学习使用IDE的跟踪调试工具，设定断点、单步跟踪、变量窗口变化等。

本人此处使用的IDE为Microsoft Visual Studio Code (不是VS)，其对应操作如下：

描述设置断点的操作： 在代码行左边单击鼠标左键，出现小红点则设置成功。再点一次小红点消失，代表取消断点。



描述逐步执行程序的操作： 按F5进入调试，程序执行到断点位置时，小红点变成一个小箭头，此时按F10可单步跳过，按F11可单步执行。



代码连接：<https://github.com/WilliamMooo/oop_course/tree/master/experiment>