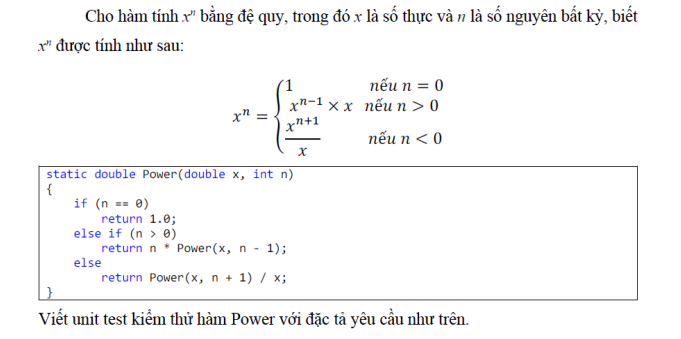
**Bài 3:** Viết unit test cho hàm Power(double x, int n)



Chỗ else if sửa return x \* Power(x, n - 1)

* Phân tích bài toán thì ta có tổng cộng 6 test case:

TC1: n = 0

TC2: n > 0

TC3: n < 0

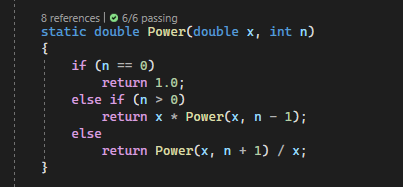
TC4: x = 0 (cơ số bằng 0)

TC5: x = ∞(cơ số bằng vô cực)

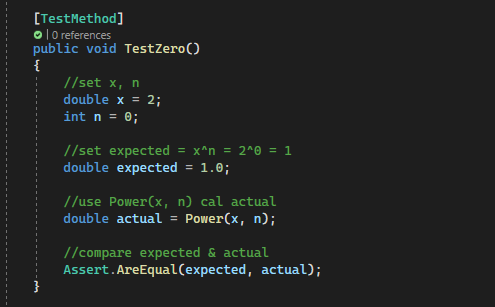
TC6: x = NaN(cơ số bằng Not a Number)

* Tiến hành viết Unit Test:

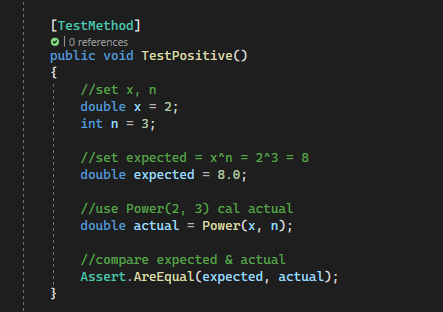
Ban đầu khởi tạo hàm Power(double x, int n) trong UnitTest1.cs cho dễ dàng test



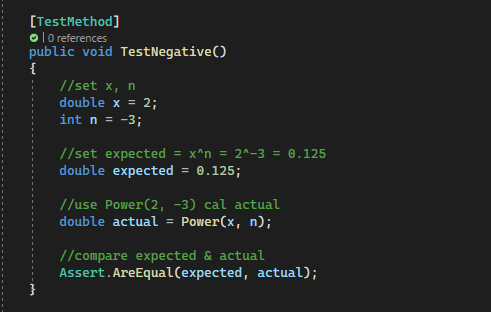
* + TC1: n = 0



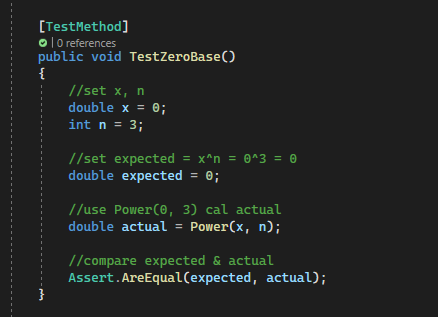
* + TC2: n > 0



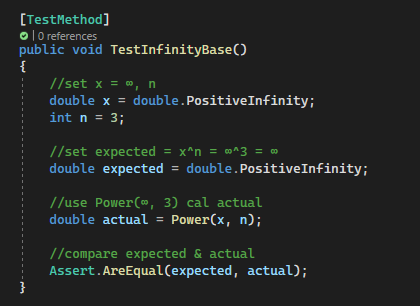
* + TC3: n < 0



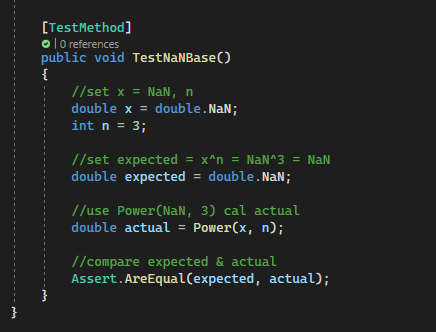
* + TC4: x = 0 (cơ số bằng 0)

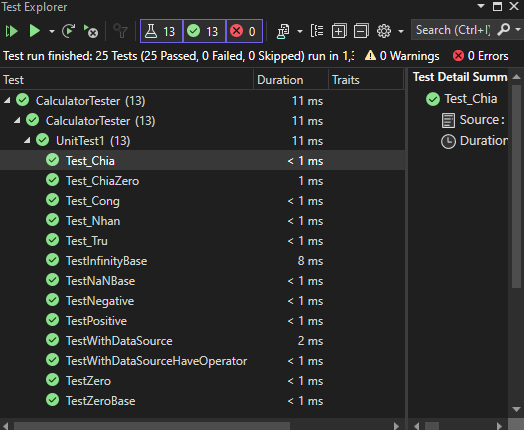


* + TC5: x = ∞(cơ số bằng vô cực)

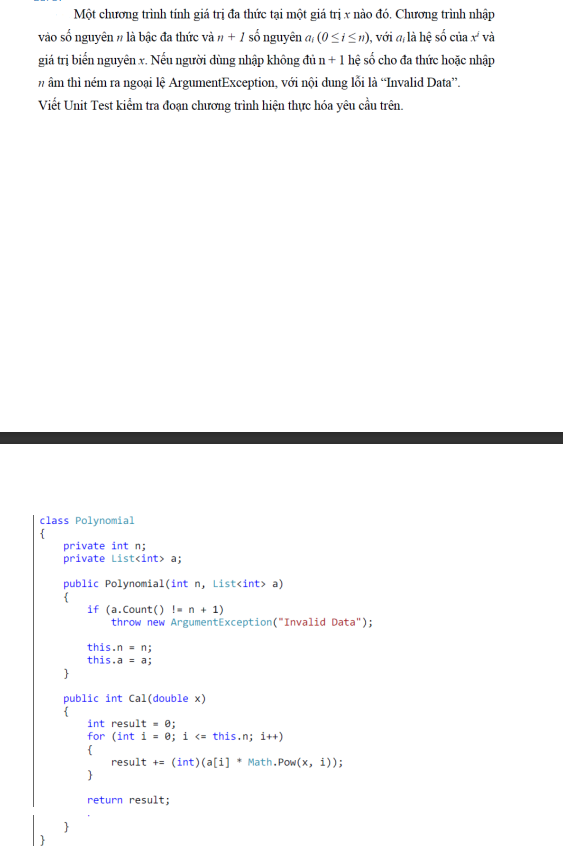


* + TC6: x = NaN(cơ số bằng Not a Number)



Tiến hành chạy Test: 

**Bài 4:** Viết Unit Test cho class đối tượng Polynomial



* Phân tích bài toán thì ta có 6 testcase:

TC1: Khởi tạo thành công (số phần tử trong list a = n + 1)

TC2: Khởi tạo thất bại(số phần tử trong list a != n + 1)

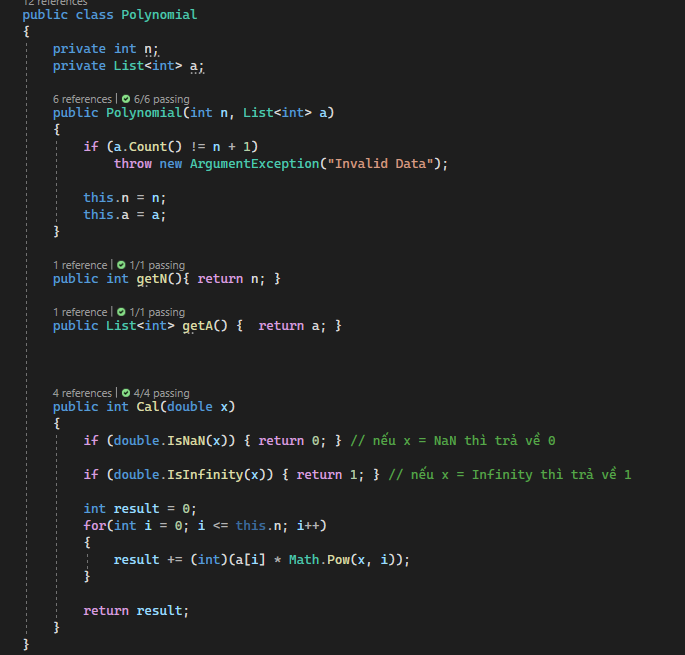
TC3: x != 0

TC4: x = 0

TC5: x = ∞

TC6: x = NaN

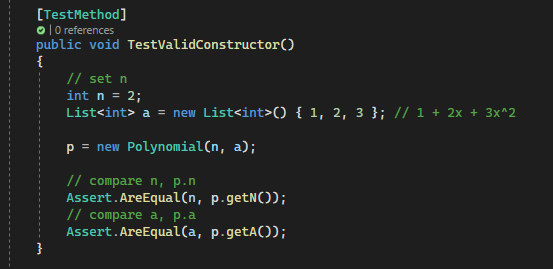
* Tiến hành viết Unit Test:
* Ban đầu, tạo class Polynomial



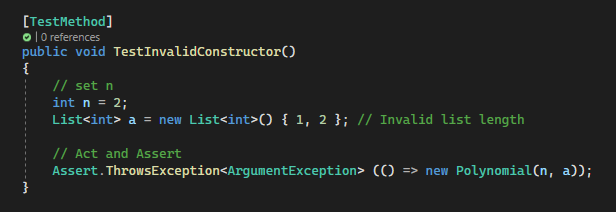
* Ở đây trong hàm *Cal(double x)* ta thêm 2 câu lệnh điều kiện để bắt 2 trường hợp x là NaN và x là vô cực
* Tiếp theo, tạo đối tượng Polynomial



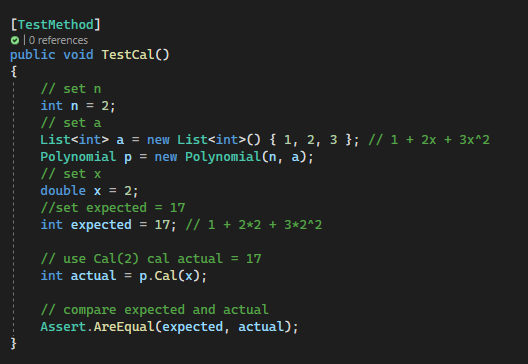
* TC1: Khởi tạo thành công (số phần tử trong list a = n + 1)



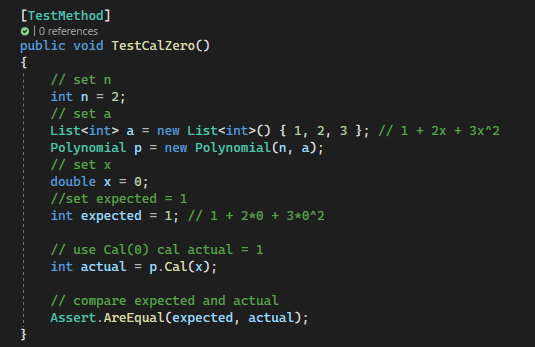
* TC2: Khởi tạo thất bại(số phần tử trong list a != n + 1)



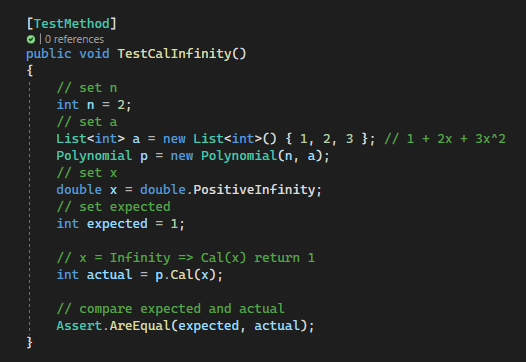
* TC3: x != 0



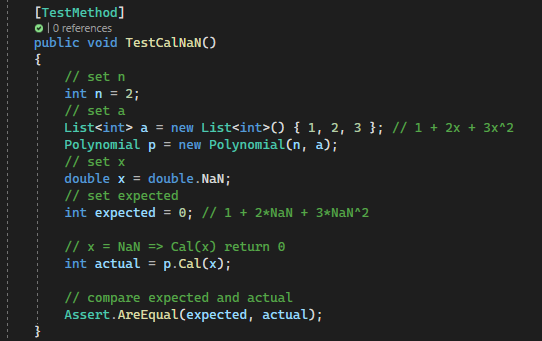
* TC4: x = 0



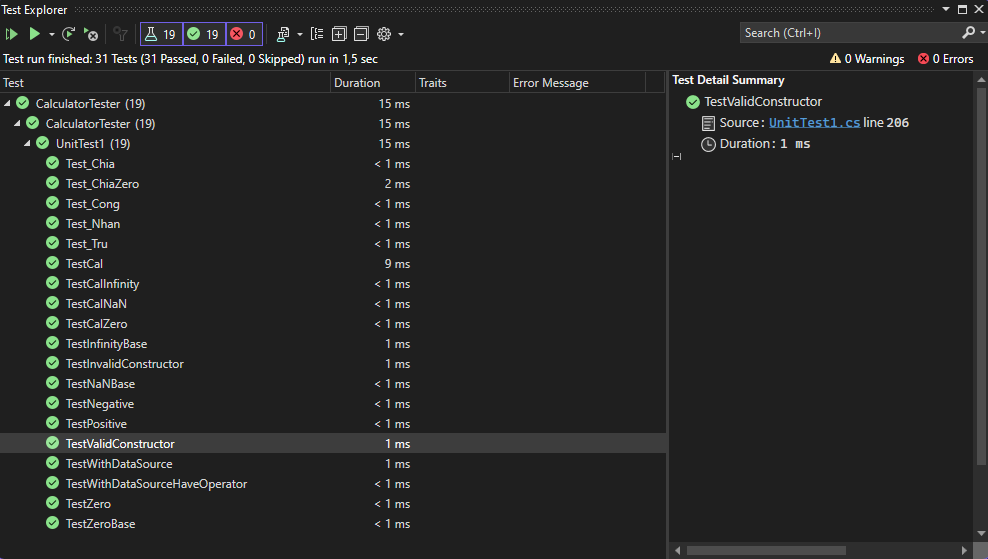
* TC5: x = ∞



* TC6: x = NaN



Tiến hành chạy Test:



**Bài 5:** Viết Unit Test cho class đối tượng Radix

* Phân tích bài toán thì ta có 4 testcase:

TC1: Khởi tạo thành công (number >= 0)

TC2: Khởi tạo thất bại(number < 0)

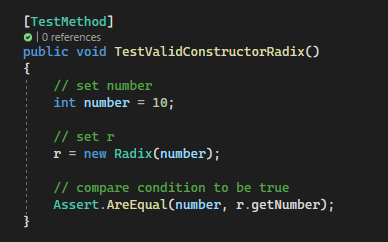
TC3: Test chuyển từ thập phân sang hệ khác(thành công)

TC4: Test chuyển từ thập phân sang hệ khác(thất bại do radix < 2 || radix > 16)

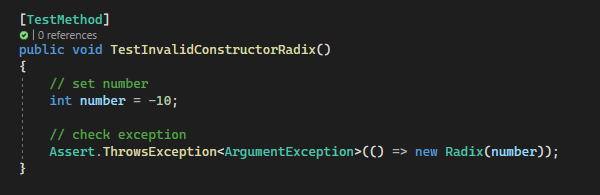
* Tiến hành viết Unit Test:
* Ban đầu khởi tạo đối tượng Radix



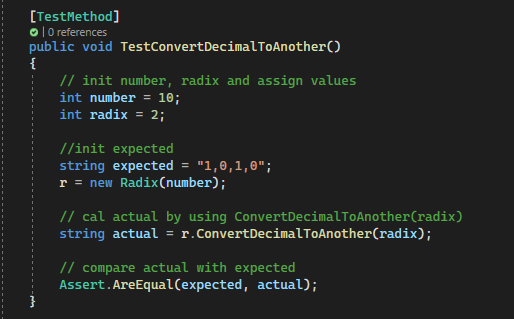
* TC1: Khởi tạo thành công (number >= 0)



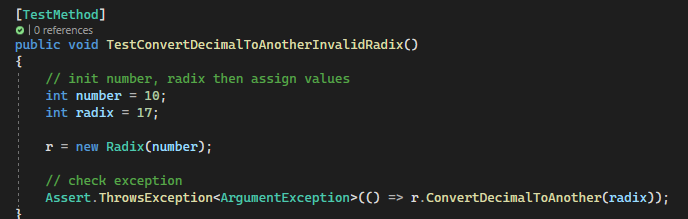
* TC2: Khởi tạo thất bại(number < 0)



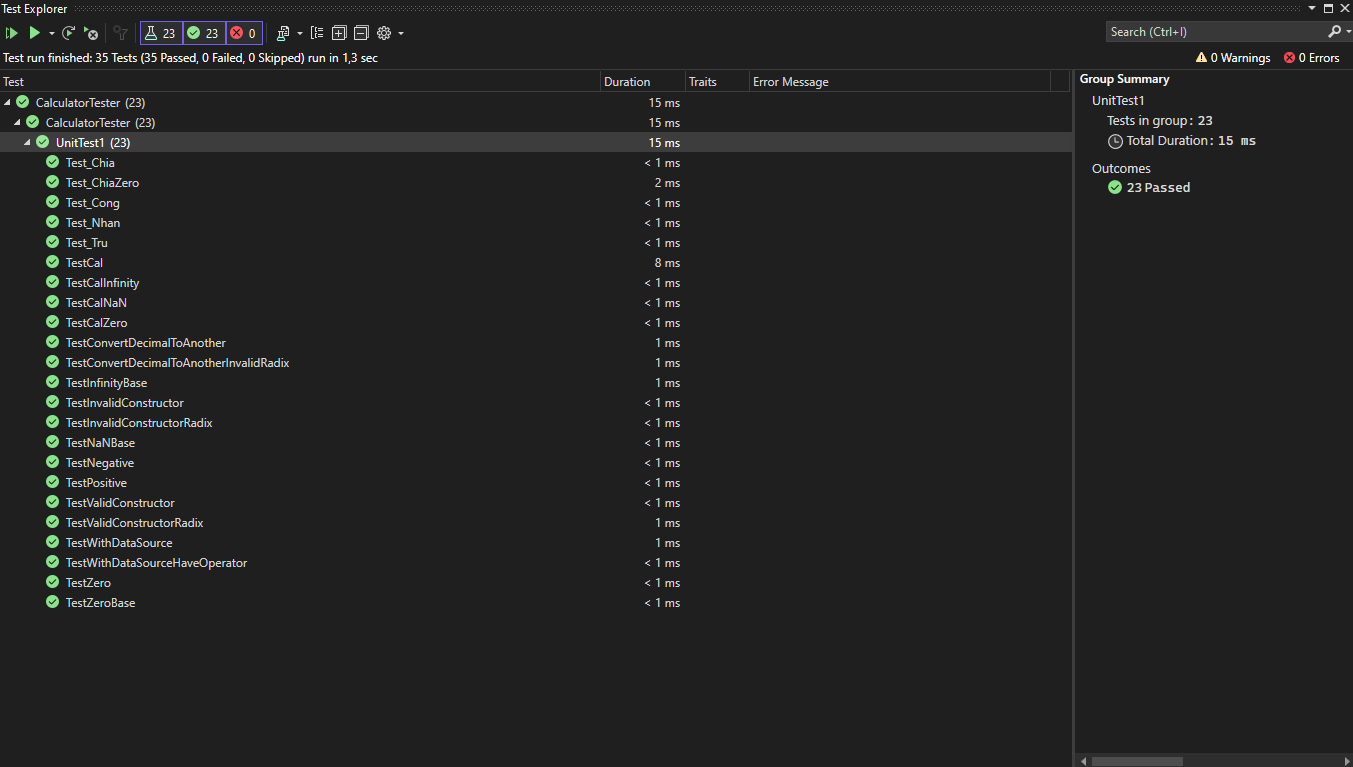
* TC3: Test chuyển từ thập phân sang hệ khác(thành công)



* TC4: Test chuyển từ thập phân sang hệ khác(thất bại do radix < 2 || radix > 16)

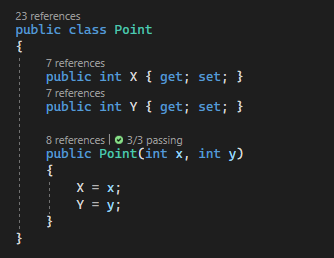


Tiến hành chạy Test:



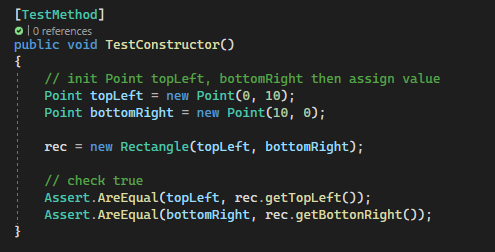
**Bài 6:**  Viết Unit Test cho đối tượng Rectangle được tạo nên từ đối tượng Point

* Phân tích bài toán ta có được 3 test case:
* TC1: Test khởi tạo thành công hình chữ nhật
* TC2: Test phương thức tính diện tích
* TC3: Test 2 hình chữ nhật giao nhau
* Tiến hành viết unit test:
* Ban đầu viết 2 đối tượng Point và Rectangle

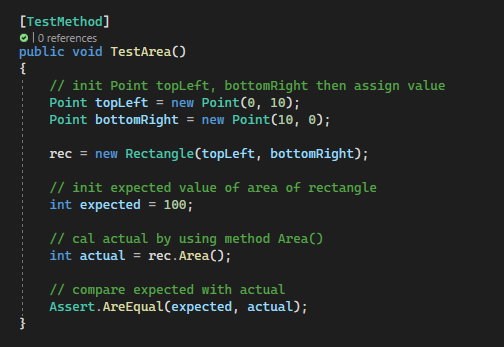




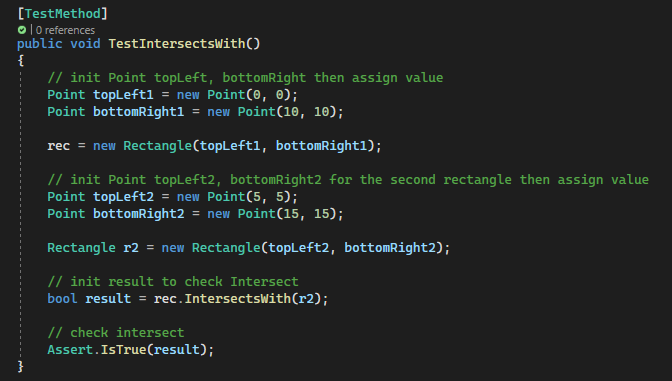
* Bước 2 khởi tạo Rec
* TC1: Test khởi tạo thành công hình chữ nhật



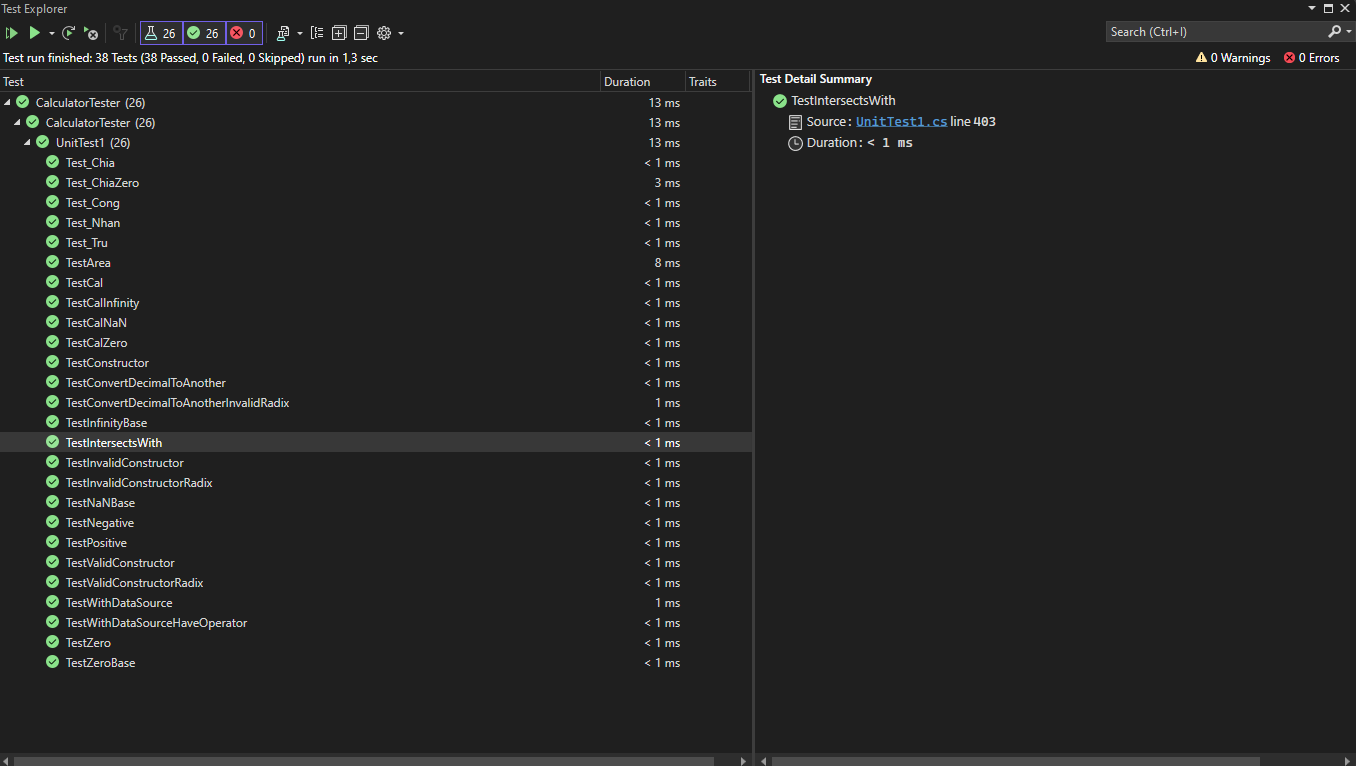
* TC2: Test phương thức tính diện tích



* TC3: Test 2 hình chữ nhật giao nhau



Tiến hành chạy test:



**Bài 7:** Viết unit test cho trung tâm gia sư

* Phân tích bài toán ta có được 4 test case:
* TC1: test phương thức tính điểm trung bình
* TC2: test phương thức có đủ điều kiện nhận học bổng không
* TC3: test phương thức hiển thị danh sách nhận học bổng
* TC4: test lấy danh sách học viên đủ điều kiện nhận học bổng
* Tiến hành viết unit test:

Viết 2 lớp đối tượng HocVien và QuanLyHocVien:

public class TrungTamGiaSu

{

public class HocVien

{

public string MaSo { get; set; }

public string HoTen { get; set; }

public string QueQuan { get; set; }

public double DiemToan { get; set; }

public double DiemLy { get; set; }

public double DiemHoa { get; set; }

public double TinhDiemTrungBinh()

{

return (DiemToan + DiemLy + DiemHoa) / 3;

}

public bool CoDuDieuKienNhanHocBong()

{

return TinhDiemTrungBinh() >= 8.0 && DiemToan >= 5.0 && DiemLy >= 5.0 && DiemHoa >= 5.0;

}

public void HienThiThongTin()

{

Console.WriteLine("Mã số: {0}", MaSo);

Console.WriteLine("Họ tên: {0}", HoTen);

Console.WriteLine("Quê quán: {0}", QueQuan);

Console.WriteLine("Điểm Toán: {0}", DiemToan);

Console.WriteLine("Điểm Lý: {0}", DiemLy);

Console.WriteLine("Điểm Hóa: {0}", DiemHoa);

Console.WriteLine("Điểm trung bình: {0}", TinhDiemTrungBinh());

Console.WriteLine("Có đủ điều kiện nhận học bổng: {0}", CoDuDieuKienNhanHocBong());

}

}

public class QuanLyHocVien

{

public string TenTrungTam { get; set; }

public List<HocVien> DanhSachHocVien { get; set; }

public QuanLyHocVien(string tenTrungTam)

{

TenTrungTam = tenTrungTam;

DanhSachHocVien = new List<HocVien>();

}

public void ThemHocVien(HocVien hocVien)

{

DanhSachHocVien.Add(hocVien);

}

public List<HocVien> LayDanhSachHocVienDuDieuKienNhanHocBong()

{

List<HocVien> ketQua = new List<HocVien>();

foreach (HocVien hocVien in DanhSachHocVien)

{

if (hocVien.CoDuDieuKienNhanHocBong())

{

ketQua.Add(hocVien);

}

}

return ketQua;

}

public void HienThiDanhSachHocVienDuDieuKienNhanHocBong()

{

Console.WriteLine("Danh sách học viên có đủ điều kiện nhận học bổng của trung tâm {0}:", TenTrungTam);

foreach (HocVien hocVien in LayDanhSachHocVienDuDieuKienNhanHocBong())

{

hocVien.HienThiThongTin();

Console.WriteLine();

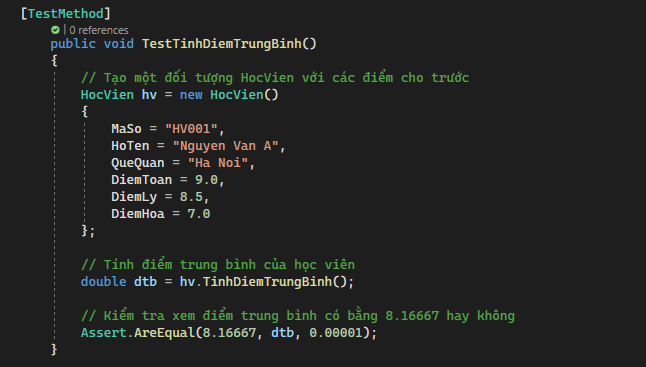
}

}

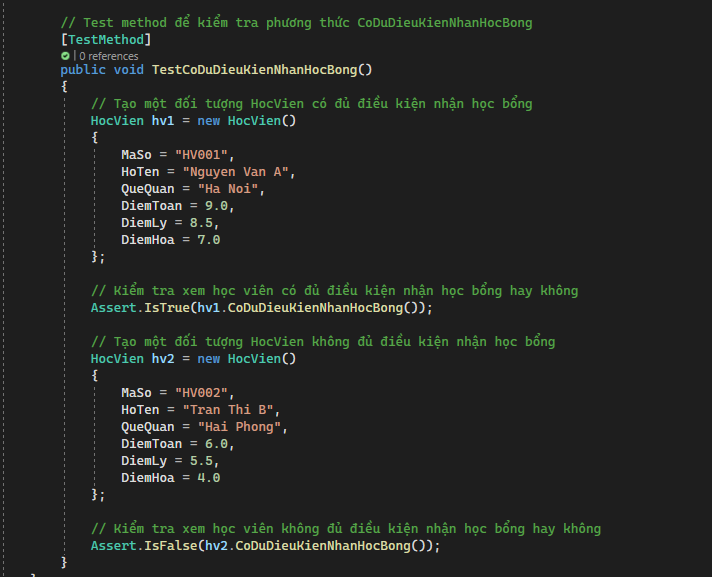
}

}

* TC1: test phương thức tính điểm trung bình



* TC2: test phương thức có đủ điều kiện nhận học bổng không



* TC3: test phương thức hiển thị danh sách nhận học bổng

// Test method để kiểm tra phương thức HienThiDanhSachHocVienDuDieuKienNhanHocBong

[TestMethod]

public void TestHienThiDanhSachHocVienDuDieuKienNhanHocBong()

{

// Tạo một đối tượng QuanLyHocVien

QuanLyHocVien qlhv = new QuanLyHocVien("ABC");

// Tạo một số đối tượng HocVien

HocVien hv1 = new HocVien()

{

MaSo = "HV001",

HoTen = "Nguyen Van A",

QueQuan = "Ha Noi",

DiemToan = 9.0,

DiemLy = 8.5,

DiemHoa = 7.0

};

HocVien hv2 = new HocVien()

{

MaSo = "HV002",

HoTen = "Tran Thi B",

QueQuan = "Hai Phong",

DiemToan = 6.0,

DiemLy = 5.5,

DiemHoa = 4.0

};

HocVien hv3 = new HocVien()

{

MaSo = "HV003",

HoTen = "Le Van C",

QueQuan = "Da Nang",

DiemToan = 8.0,

DiemLy = 8.0,

DiemHoa = 8.0

};

// Thêm các đối tượng HocVien vào QuanLyHocVien

qlhv.ThemHocVien(hv1);

qlhv.ThemHocVien(hv2);

qlhv.ThemHocVien(hv3);

// Tạo một đối tượng StringWriter để chứa kết quả hiển thị

System.IO.StringWriter sw = new System.IO.StringWriter();

// Đổi luồng xuất chuẩn sang StringWriter

System.Console.SetOut(sw);

// Gọi phương thức HienThiDanhSachHocVienDuDieuKienNhanHocBong

qlhv.HienThiDanhSachHocVienDuDieuKienNhanHocBong();

// Lấy kết quả hiển thị từ StringWriter

string result = sw.ToString();

// Kiểm tra xem kết quả có chứa đúng thông tin của hv1 và hv3 hay không

Assert.IsTrue(result.Contains("HV001"));

Assert.IsTrue(result.Contains("Nguyen Van A"));

Assert.IsTrue(result.Contains("HV003"));

Assert.IsTrue(result.Contains("Le Van C"));

}

* TC4: test lấy danh sách học viên đủ điều kiện nhận học bổng

[TestMethod]

public void TestLayDanhSachHocVienDuDieuKienNhanHocBong()

{

// Tạo một đối tượng QuanLyHocVien

QuanLyHocVien qlhv = new QuanLyHocVien("ABC");

// Tạo một số đối tượng HocVien

HocVien hv1 = new HocVien()

{

MaSo = "HV001",

HoTen = "Nguyen Van A",

QueQuan = "Ha Noi",

DiemToan = 9.0,

DiemLy = 8.5,

DiemHoa = 7.0

};

HocVien hv2 = new HocVien()

{

MaSo = "HV002",

HoTen = "Tran Thi B",

QueQuan = "Hai Phong",

DiemToan = 6.0,

DiemLy = 5.5,

DiemHoa = 4.0

};

HocVien hv3 = new HocVien()

{

MaSo = "HV003",

HoTen = "Le Van C",

QueQuan = "Da Nang",

DiemToan = 8.0,

DiemLy = 8.0,

DiemHoa = 8.0

};

// Thêm các đối tượng HocVien vào QuanLyHocVien

qlhv.ThemHocVien(hv1);

qlhv.ThemHocVien(hv2);

qlhv.ThemHocVien(hv3);

// Lấy danh sách học viên có đủ điều kiện nhận học bổng

List<HocVien> ds = qlhv.LayDanhSachHocVienDuDieuKienNhanHocBong();

// Kiểm tra xem danh sách có chứa đúng 2 học viên hv1 và hv3 hay không

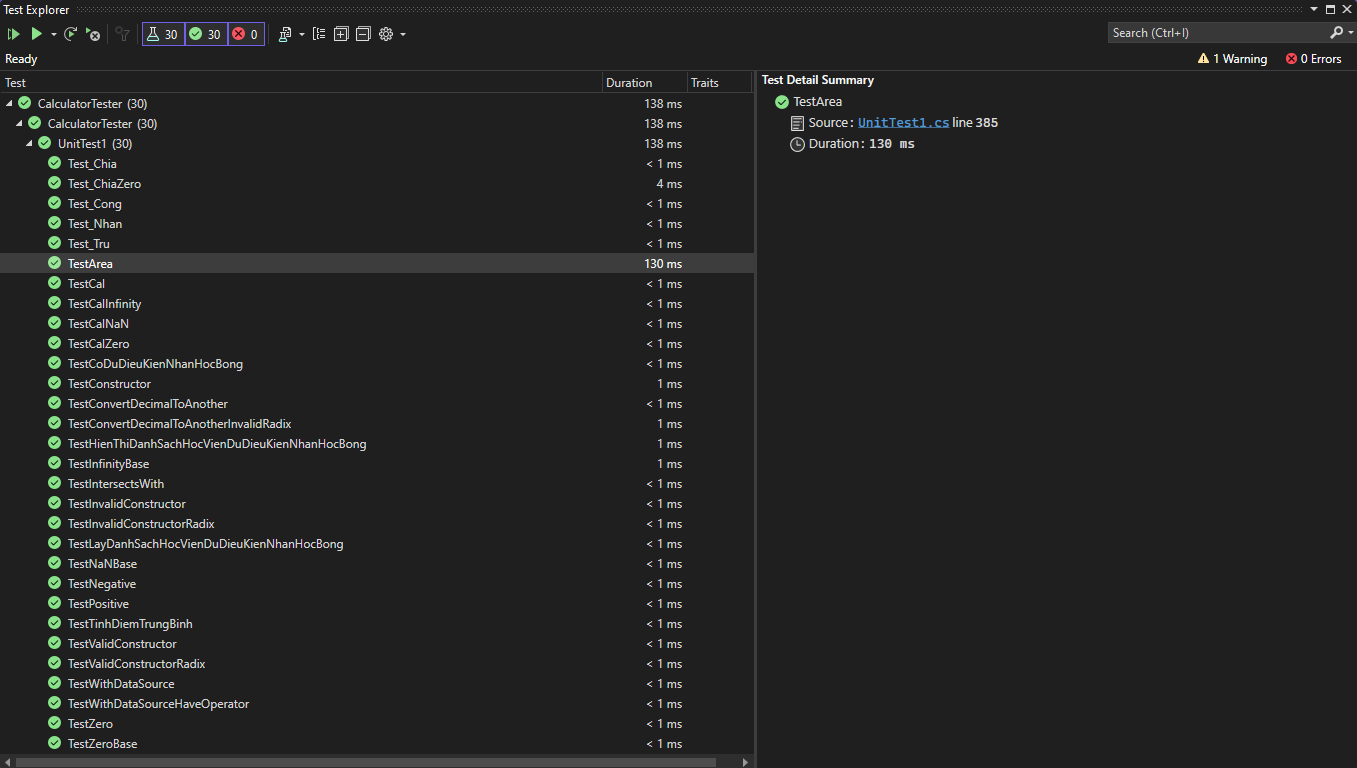
Assert.AreEqual(2, ds.Count);

Assert.IsTrue(ds.Contains(hv1));

Assert.IsTrue(ds.Contains(hv3));

}

* Tiến hành chạy test:



Link github cho cô check: https://github.com/nhantt71/demo\_KTPM.git