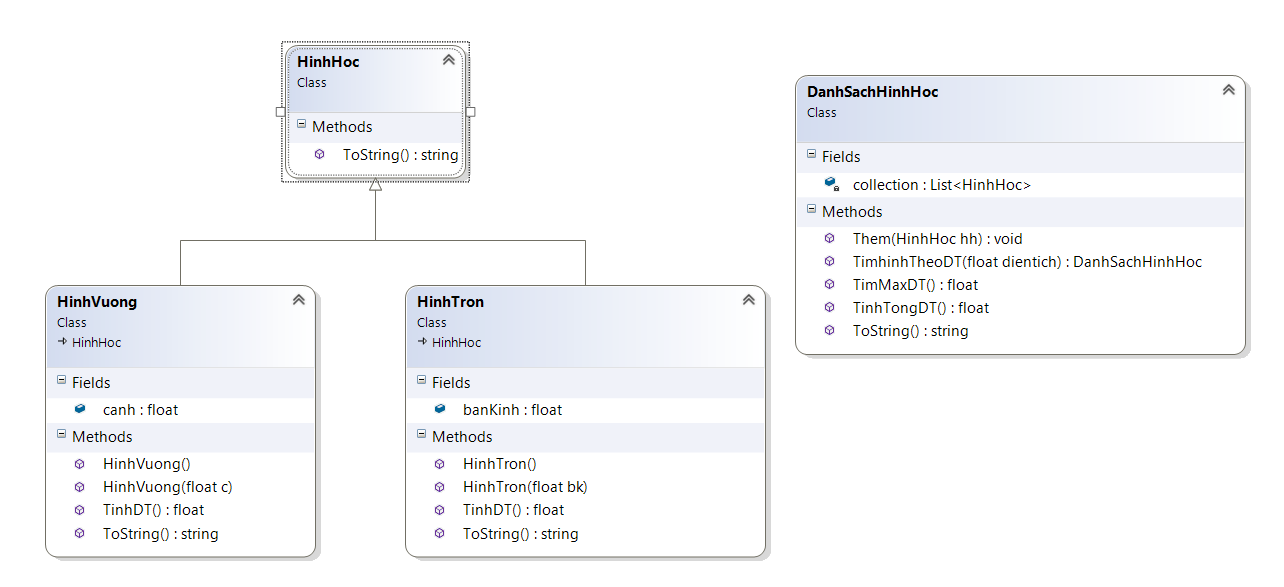
**Lab 7: Tính kế thừa**

Bài 1: Sử dụng dự án 6\_TinhKeThua trong Code mẫu

* Đóng mở các chú thích trong dự án để chạy và kiểm tra kết quả
* Giải thích các dòng lệnh trong dự án và kết quả thực hiện chương trình:
* Tạo 1 lớp cha với 3 phương thức, 1 phương thức không truyền vào tham số, 1 phương thức truyền vào 2 tham số là a và b, 1 phương thức để xuất lớp cha ra màn hình là 1 số nguyên và 1 chuỗi bất kì
* Tạo ra 1 lớp con kế thừa từ lớp cha nên sẽ có các phương thức mà lớp cha cho phép (XuatCha) và thêm 3 phương thức riêng của con - 1 phương thức không truyền vào tham số, 1 phương thức truyền vào 3 tham số là a,b,c với a,b dựa vào lớp Cha nên khi truyền tham số ta chỉ cần truyền cho z=c vì a,b đã dựa vào lớp cha , 1 phương thức để xuất lớp con ra màn hình là 1 số nguyên và 1 chuỗi bất kì và 1 số nguyên - - ở program khai báo 1 biến cha thuộc kiểu dữ liệu Cha và truyền cho nó dữ liệu là “1” và “Xin chao”. Sau đó dùng phương thức cha.XuatCha(); nên sẽ xuất ra màn hình Lop cha x= 1, y= Xin chao sau đó khai báo 1 biến c thuộc kiểu dữ liệu LopCon và truyền cho nó dữ liệu là “1”,”Hello”,”2” sau đó dùng phương thức c.XuatCha nên c chỉ xuất ra màn hinh 2 tham số la “1” và “Hello” vì c kế thừa phương thức XuatCha cua lop Cha nên chỉ có thể xuất x và y ra ngoài nên kết quả chỉ là Lop cha x= 1, y= Hello sau đó ta lại sử dụng lệnh gán cha=c và không thể làm ngược lại là c=cha vì lớp con kế thừa từ cha nên con có thể gán cho cha vì không thể chuyển từ kiểu cha sang kiểu con được vì kiểu con sẽ lớn hơn kiểu cha tiếp tục lại gán c cho o thuộc kiểu dữ liệu Object được vì lớp Object là cha của tất cả các lớp vì object là cha của tất cả của cấc lớp nên có hể gán cha cho o được và nếu làm ngược lại thì sẽ không được, sau đó ta dùng phương thức cha.XuatCha thì màn hình sẽ xuất ra màn hình Lop cha x= 1, y= Hello vì lúc này con đã gán cho cha nên dữ liệu của con sẽ được thay cho cha nên mới xuất ra màn hình Lop cha x= 1, y= Hello, sau đó sử dụng phương thức c.XuatCon thì sẽ xuat ra màn hình Con x= 1, y = Hello, z= 2 vì đây là phương thức riêng của lớp con nên sẽ đầy đủ 3 thành phần x,y,z còn ở dòng lệnh cha.XuatCha thì chỉ có 2 thành phần x và y mà x và y này lại mang dữ liệu của lớp con vì con đã gán cho cha qua dòng lệnh cha=c

/-hea

Bài 2: Sử dụng dự án 7\_DuAnHinhHoc và thực hiện các yêu cầu sau:



class DanhSachHinhHoc

{

List<HinhHoc> collection = new List<HinhHoc>();

public void Them(HinhHoc hh)

{

collection.Add(hh);

}

public float TinhTongDT()

{

float tong = 0;

foreach (var item in collection)

{

if (item is HinhTron)

tong += ((HinhTron)item).TinhDT();

if (item is HinhVuong)

tong += ((HinhVuong)item).TinhDT();

}

return tong;

}

public float TimMaxDT()

{

float max = -1;

foreach (var item in collection)

{

float dt = 0;

if (item is HinhTron)

dt = ((HinhTron)item).TinhDT();

if (item is HinhVuong)

dt = ((HinhVuong)item).TinhDT();

if (max < dt)

max = dt;

}

return max;

}

public DanhSachHinhHoc TimhinhTheoDT(float dientich)

{

DanhSachHinhHoc kq = new DanhSachHinhHoc();

foreach (var item in collection)

{

float dt = 0;

if (item is HinhTron)

dt = ((HinhTron)item).TinhDT();

if (item is HinhVuong)

dt = ((HinhVuong)item).TinhDT();

if (dt == dientich)

kq.Them(item);

}

return kq;

}

public override string ToString()

{

string s = "";

foreach (var item in collection)

{

s += "\n" + item;

}

return s;

}

}

Trong lớp DanhSachHinhHoc thực hiện các yêu cầu sau:

1. Bổ sung lớp hình chữ nhật

class HinhChuNhat : HinhHoc

{

public float ChieuDai { get; set; }

public float ChieuRong { get; set; }

public HinhChuNhat(float chieuDai, float chieuRong):base(chieuDai)

{

ChieuDai = chieuDai;

ChieuRong = chieuRong;

}

public override float TinhDienTich()

{

return ChieuDai \* ChieuRong;

}

public override string ToString()

{

return $"Hinh chu nhat co chieu dai {ChieuDai}, chieu rong {ChieuRong}, dien tich la {TinhDienTich()}";

}

}

1. Nhập từ File

public void DocTuFile(string filename)

{

StreamReader sr = new StreamReader(filename);

string line;

while ((line = sr.ReadLine()) != null)

{

var data = line.Split(',');

if (data.Length > 0)

{

float a, b, c;

switch (data[0])

{

case "HT":

a = float.Parse(data[1]);

Them(new HinhTron(a));

break;

case "HV":

b = float.Parse(data[1]);

Them(new HinhVuong(b));

break;

case "HCN":

a = float.Parse(data[1]);

b = float.Parse(data[2]);

Them(new HinhChuNhat(a, b));

break;

default:

break;

}

}

}

1. Tìm hình vuông có cạnh lớn nhất, nhỏ nhất

public QuanLyHinhHoc TimHinh(Func<HinhHoc, bool> dk)

{

QuanLyHinhHoc kq = new QuanLyHinhHoc();

foreach (HinhHoc hinhHoc in dsHH)

{

if (dk(hinhHoc))

{

kq.Add(hinhHoc);

}

}

return kq;

}

public float Max(Func<HinhHoc, float> giatri)

{

if (SoLuong > 0)

{

float max = giatri(this[0]);

for (int i = 1; i < SoLuong; i++)

{

if (max < giatri(this[i]))

max = giatri(this[i]);

}

return max;

}

return 0;

}

public float Min(Func<HinhHoc, float> giatri)

{

if (SoLuong > 0)

{

float min = giatri(this[0]);

for (int i = 1; i < SoLuong; i++)

{

if (min > giatri(this[i]))

min = giatri(this[i]);

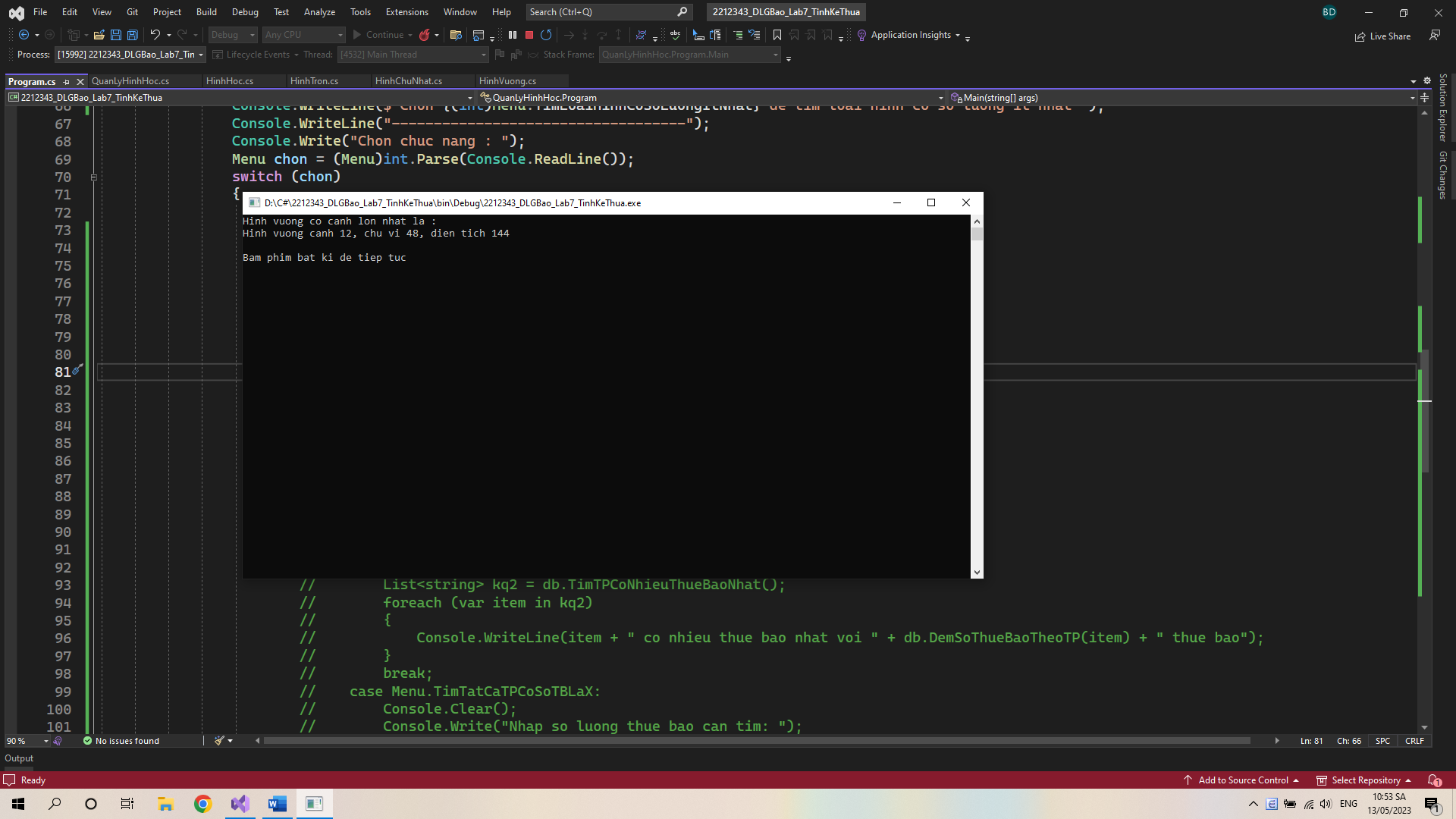
}

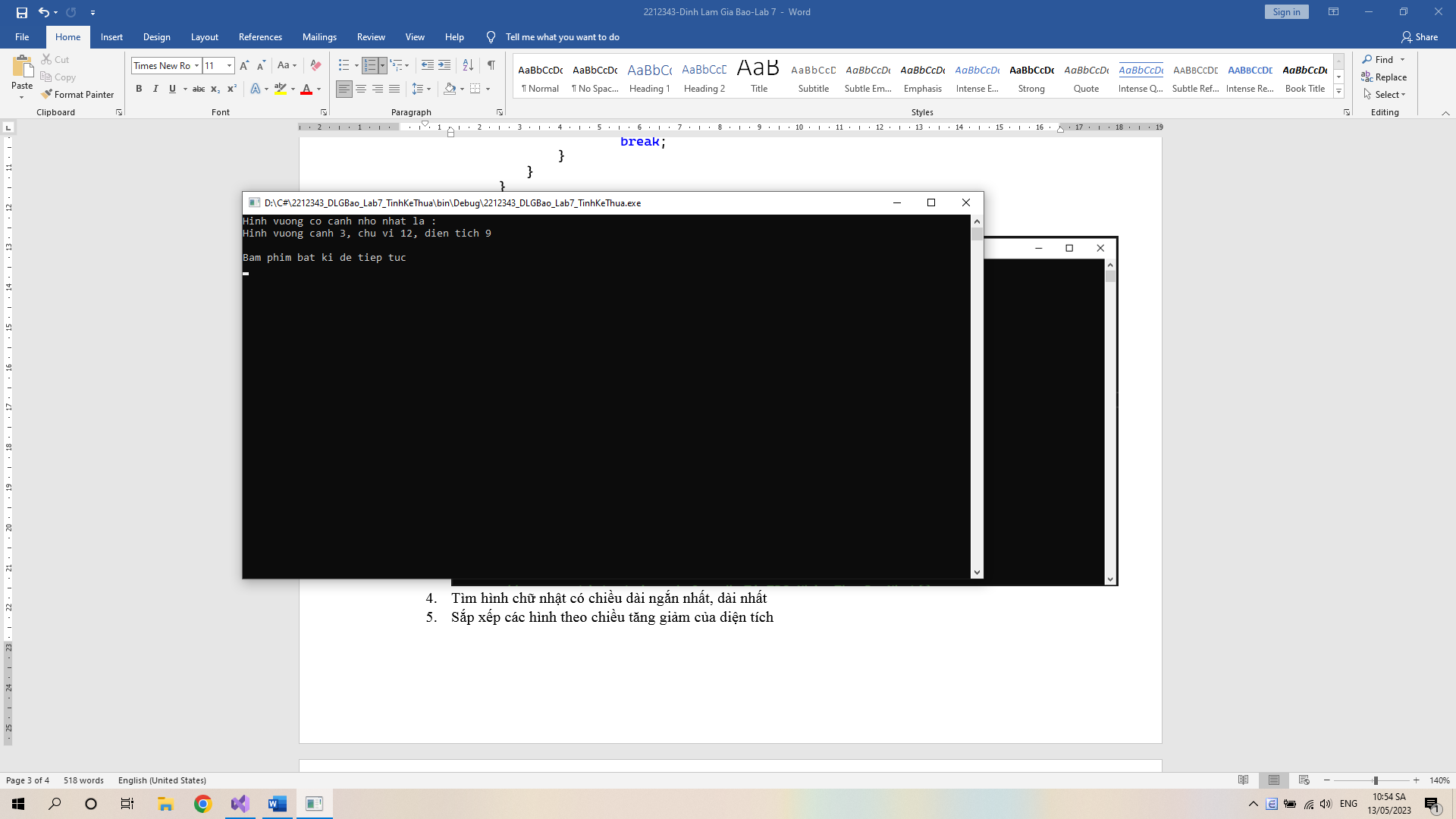
return min;

}

return 0;

}





1. Tìm hình chữ nhật có chiều dài ngắn nhất, dài nhất

case Menu.TimHCNCoChieuDaiNganNhat:

Console.Clear();

Console.WriteLine(qlhh.ToString());

dshcn = qlhh.TimHinh(hh => hh is HinhChuNhat);

min = dshcn.Min(hcn => ((HinhChuNhat)hcn).ChieuDai);

Console.WriteLine("Hinh chu nhat co chieu dai ngan nhat la : ");

Console.WriteLine(dshcn.TimHinh(h => ((HinhChuNhat)h).ChieuDai == min));

break;

case Menu.TimHCNCoChieuDaiDaiNhat:

Console.Clear();

Console.WriteLine(qlhh.ToString());

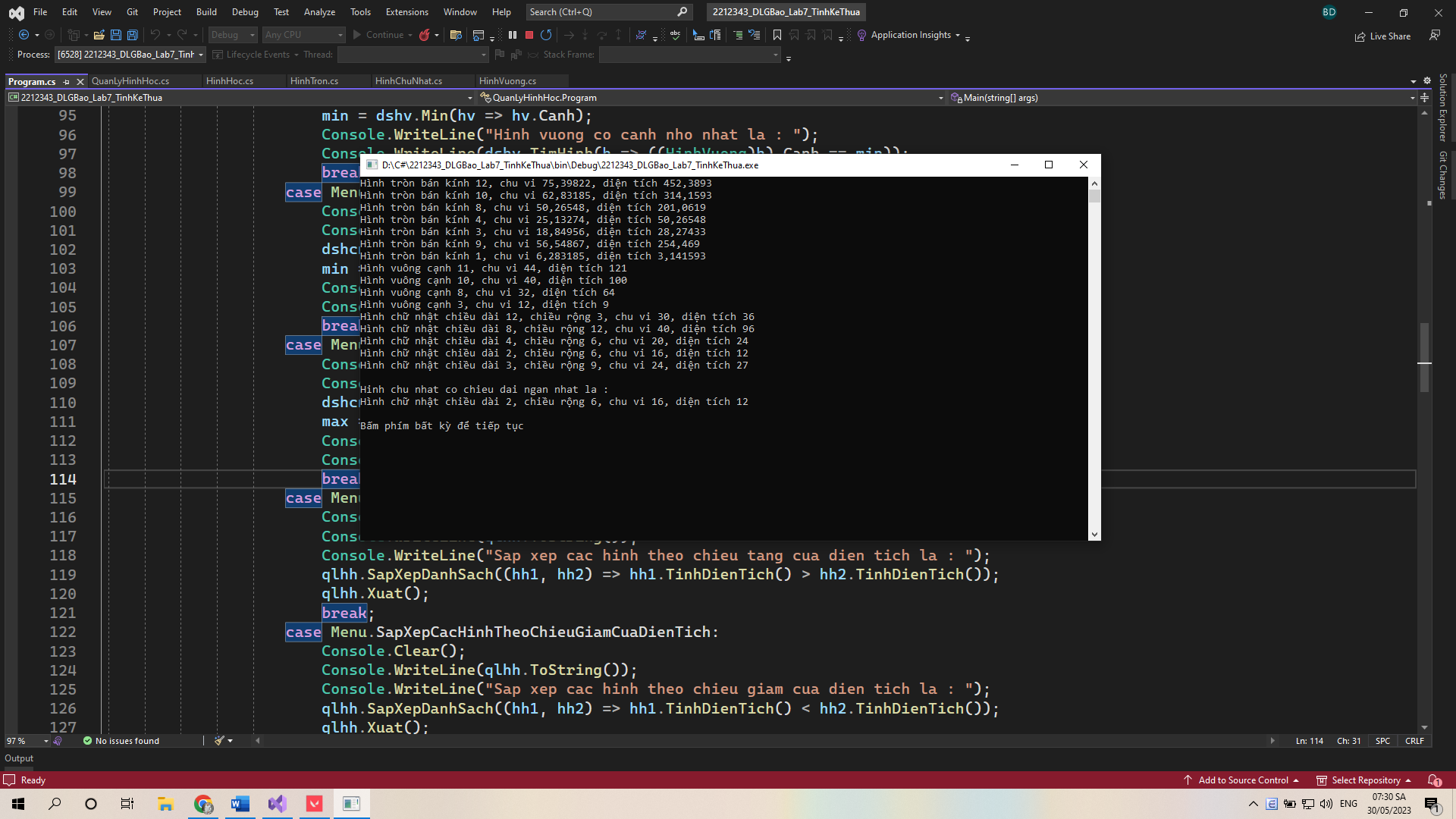
dshcn = qlhh.TimHinh(hh => hh is HinhChuNhat);

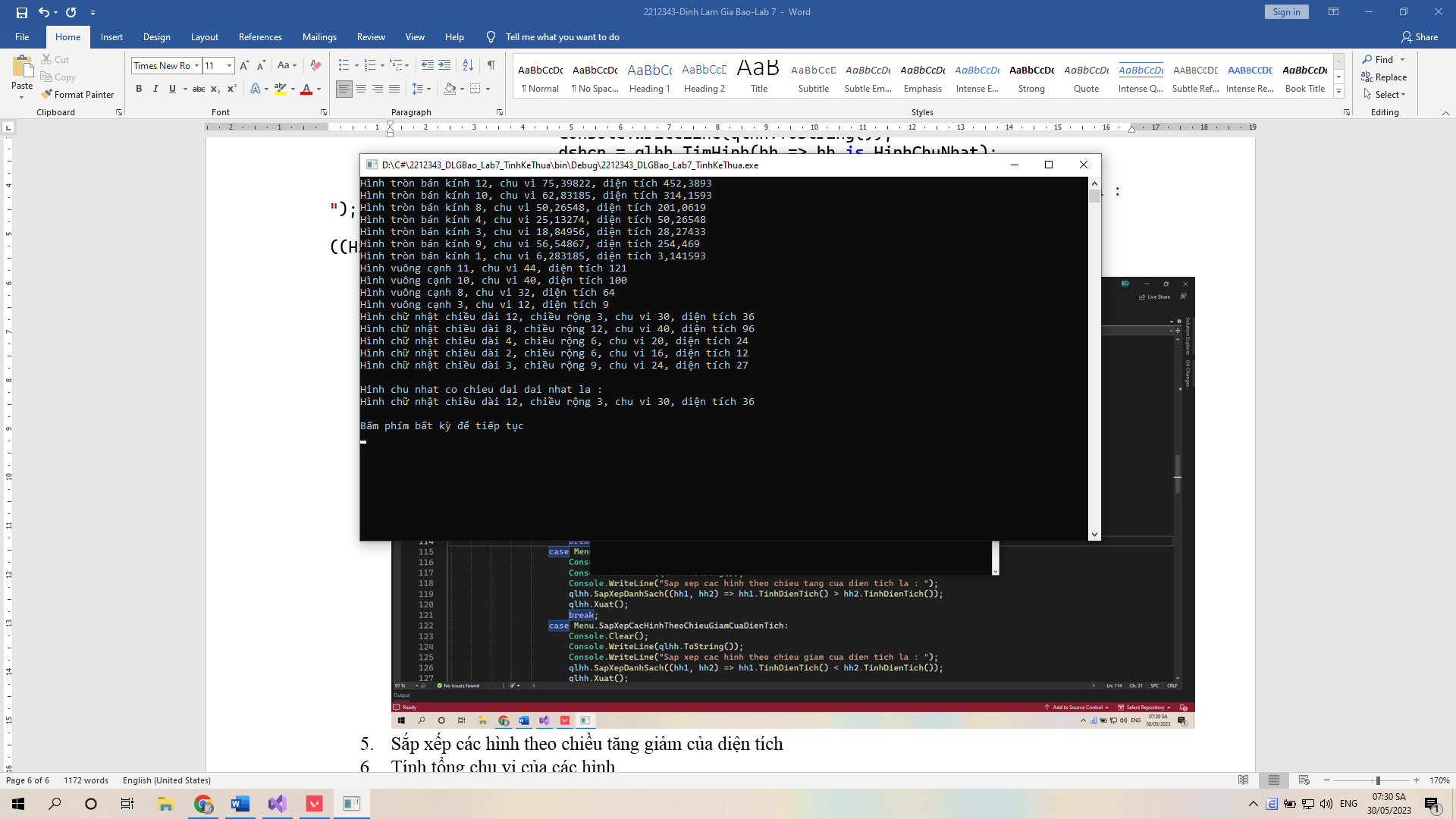
max = dshcn.Max(hcn => ((HinhChuNhat)hcn).ChieuDai);

Console.WriteLine("Hinh chu nhat co chieu dai dai nhat la : ");

Console.WriteLine(dshcn.TimHinh(h => ((HinhChuNhat)h).ChieuDai == max));

break;





1. Sắp xếp các hình theo chiều tăng giảm của diện tích

case Menu.SapXepCacHinhTheoChieuTangCuaDienTich:

Console.Clear();

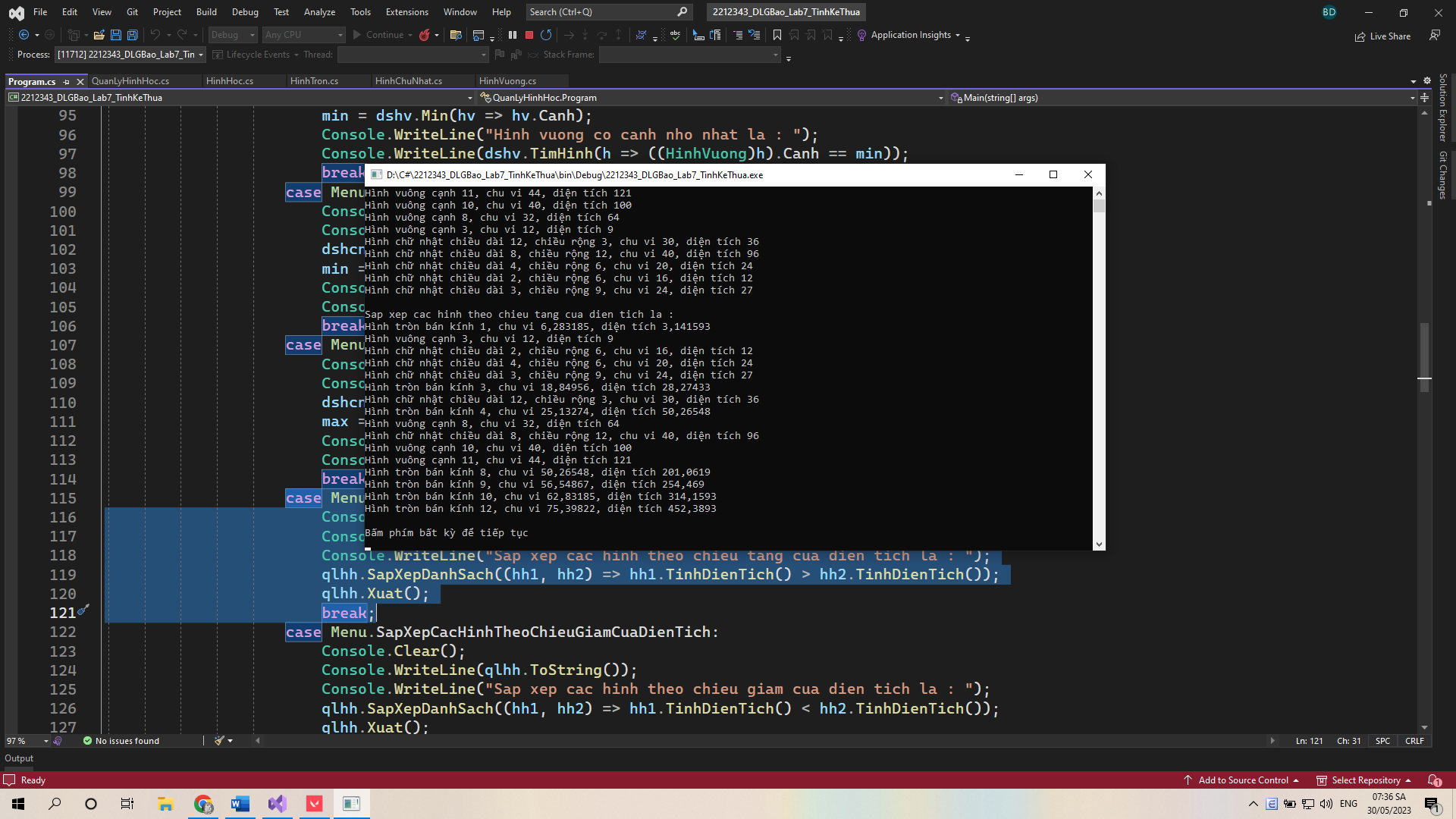
Console.WriteLine(qlhh.ToString());

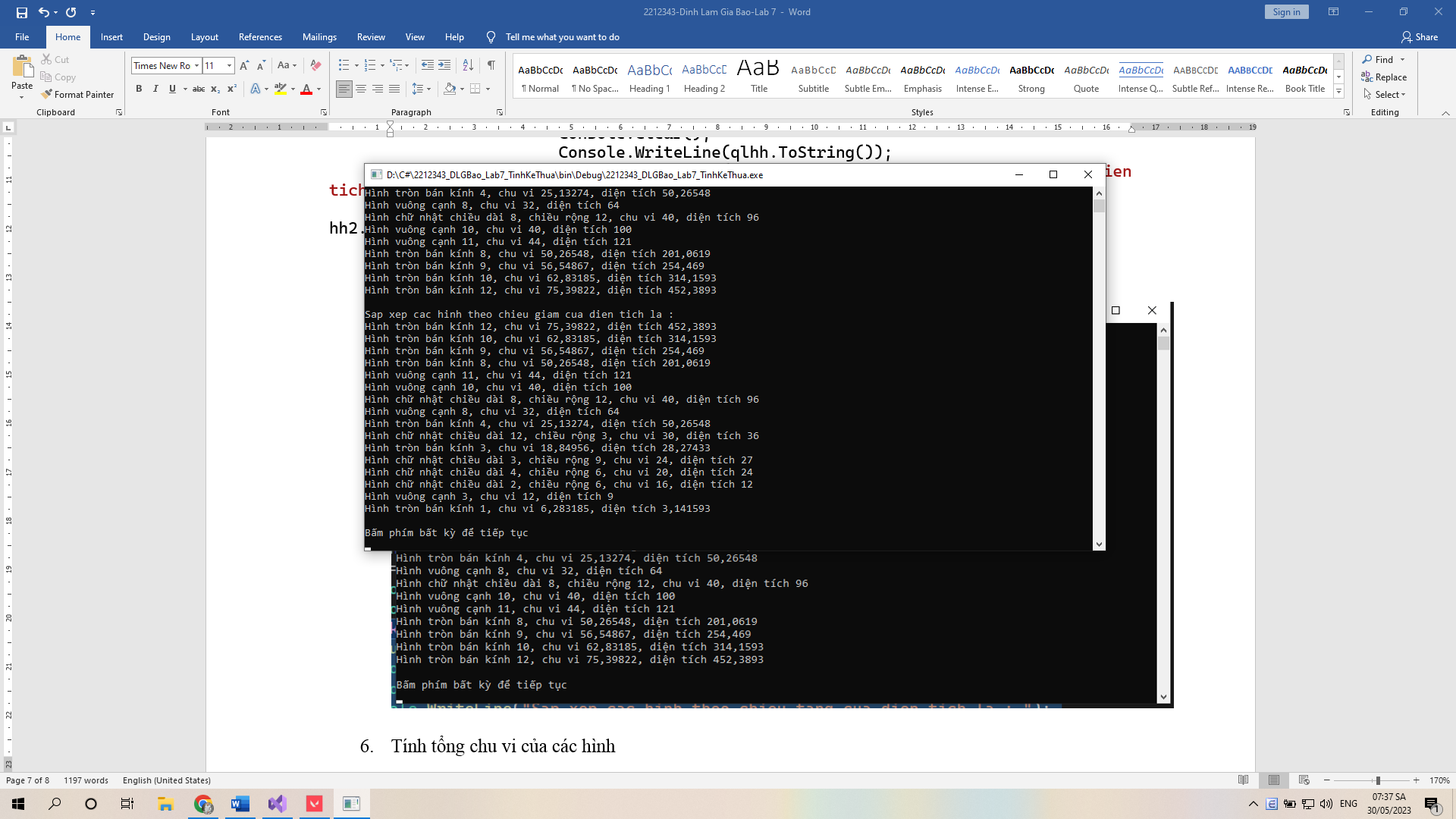
Console.WriteLine("Sap xep cac hinh theo chieu tang cua dien tich la : ");

qlhh.SapXepDanhSach((hh1, hh2) => hh1.TinhDienTich() > hh2.TinhDienTich());

qlhh.Xuat();

break;





1. Tính tổng chu vi của các hình

case Menu.TongChuViCacHinh:

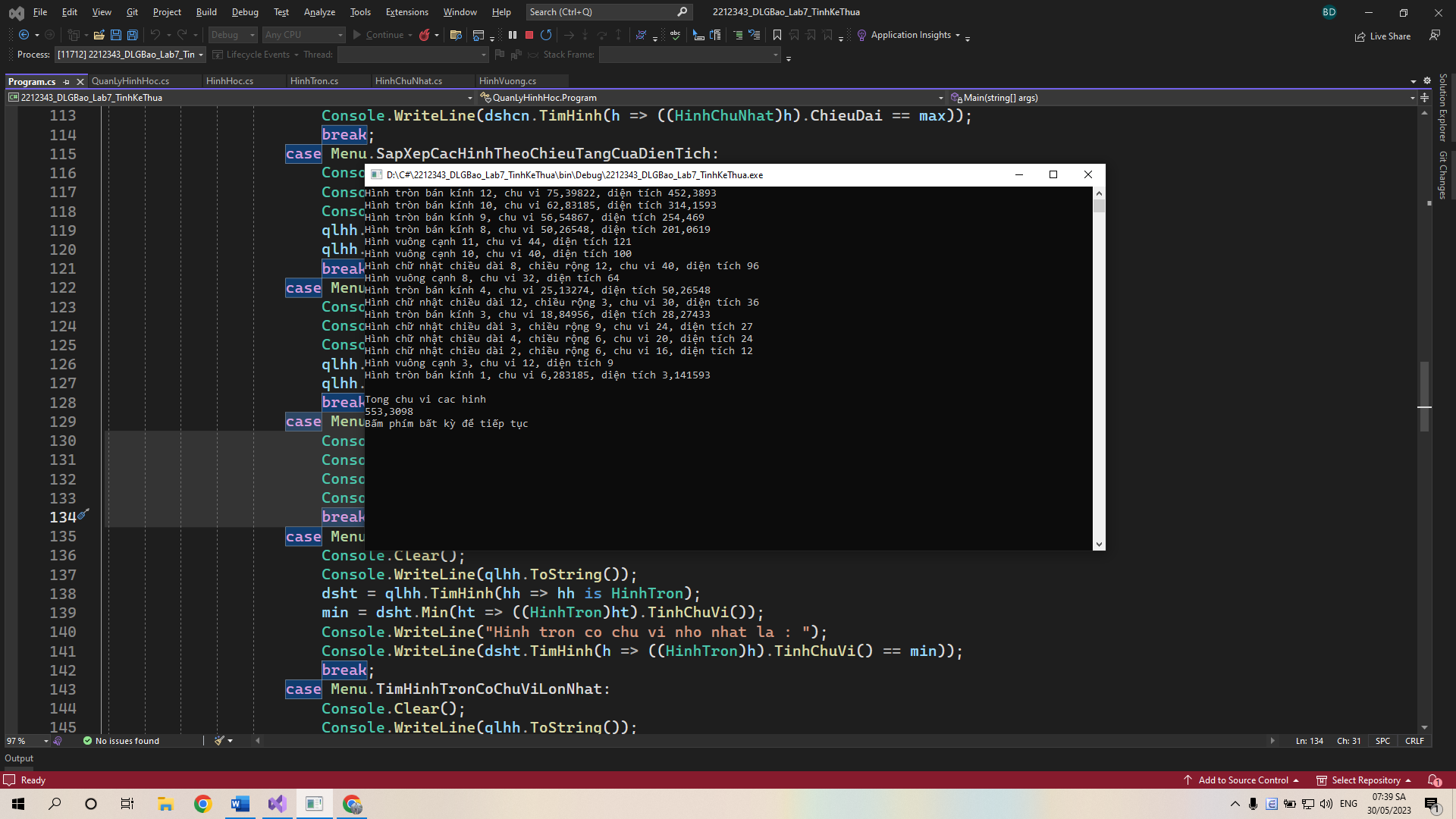
Console.Clear();

Console.WriteLine(qlhh.ToString());

Console.WriteLine("Tong chu vi cac hinh");

Console.WriteLine(qlhh.Tong((hh) => hh.TinhChuVi()));

break;



1. Tìm hình tròn có chu vi nhỏ nhất, lớn nhất

case Menu.TimHinhTronCoChuViNhoNhat:

Console.Clear();

Console.WriteLine(qlhh.ToString());

dsht = qlhh.TimHinh(hh => hh is HinhTron);

min = dsht.Min(ht => ((HinhTron)ht).TinhChuVi());

Console.WriteLine("Hinh tron co chu vi nho nhat la : ");

Console.WriteLine(dsht.TimHinh(h => ((HinhTron)h).TinhChuVi() == min));

break;

case Menu.TimHinhTronCoChuViLonNhat:

Console.Clear();

Console.WriteLine(qlhh.ToString());

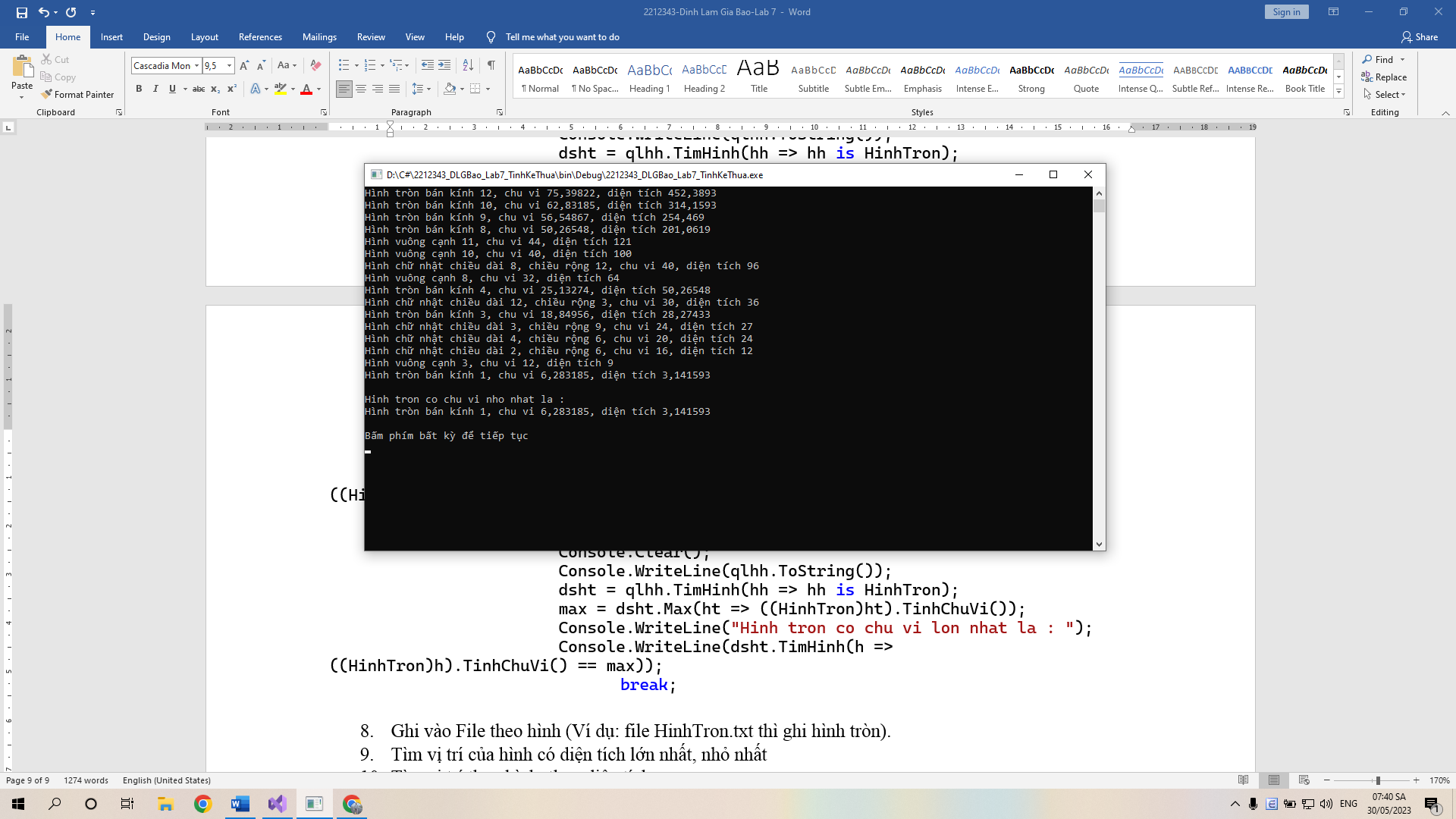
dsht = qlhh.TimHinh(hh => hh is HinhTron);

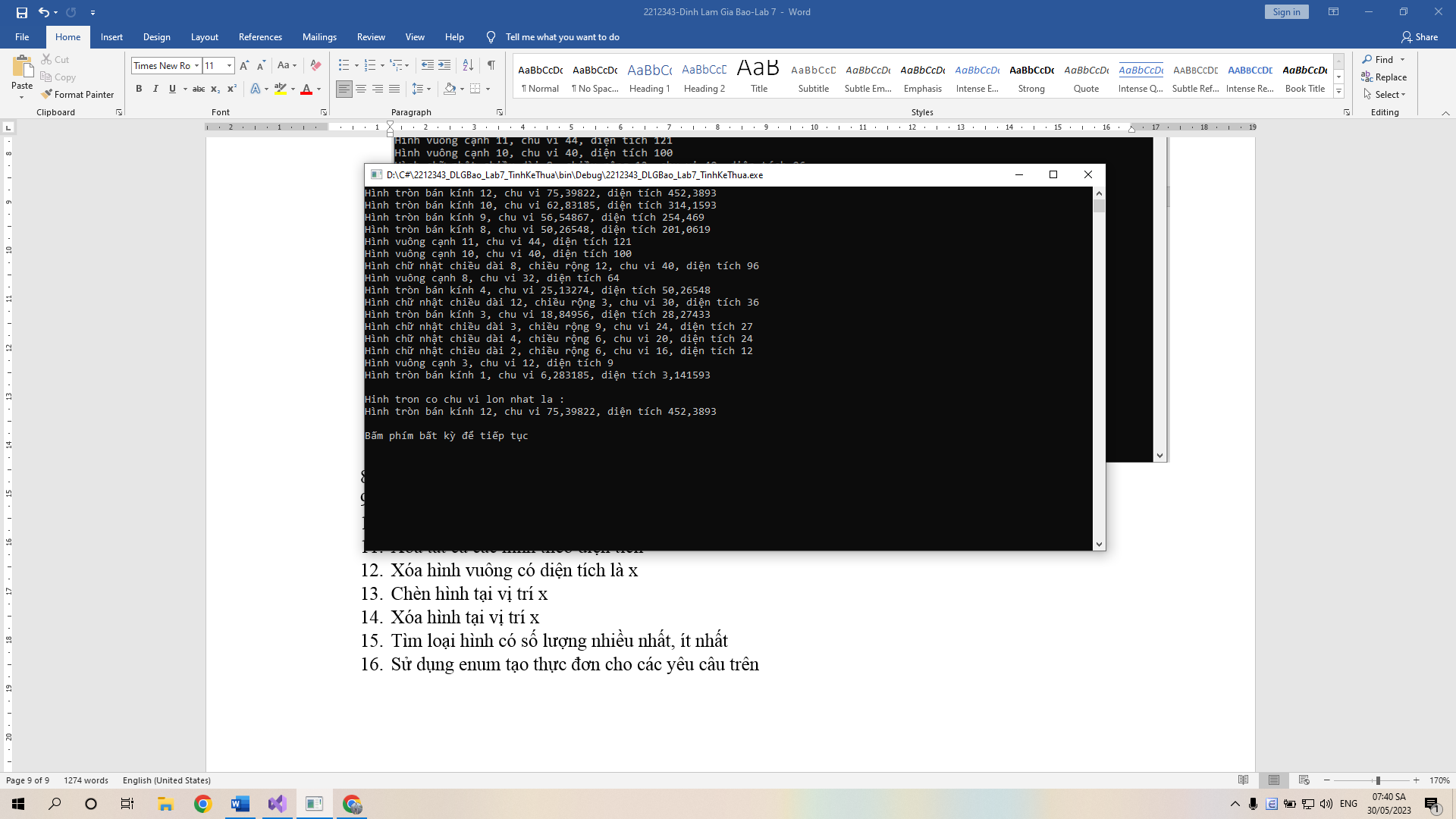
max = dsht.Max(ht => ((HinhTron)ht).TinhChuVi());

Console.WriteLine("Hinh tron co chu vi lon nhat la : ");

Console.WriteLine(dsht.TimHinh(h => ((HinhTron)h).TinhChuVi() == max));

break;





1. Ghi vào File theo hình (Ví dụ: file HinhTron.txt thì ghi hình tròn).
2. Tìm vị trí của hình có diện tích lớn nhất, nhỏ nhất

case Menu.TimViTriHinhCoDienTichLonNhat:

Console.Clear();

Console.WriteLine(qlhh.ToString());

max = qlhh.Max(hh => ((HinhHoc)hh).TinhDienTich());

List<int> kq2 = new List<int>();

kq2 = qlhh.TimViTriHinh(h => h.TinhDienTich() == max);

Console.Write($"Vị trí của hình có diện tích lớn nhất là: ");

foreach (int i in kq2)

{

Console.WriteLine(i + "\t");

}

break;

case Menu.TimViTriHinhCoDienTichNhoNhat:

Console.Clear();

Console.WriteLine(qlhh.ToString());

min = qlhh.Min(hh => ((HinhHoc)hh).TinhDienTich());

List<int> kq3 = new List<int>();

kq3 = qlhh.TimViTriHinh(h => h.TinhDienTich() == min);

Console.Write($"Vị trí của hình có diện tích nhỏ nhất là: ");

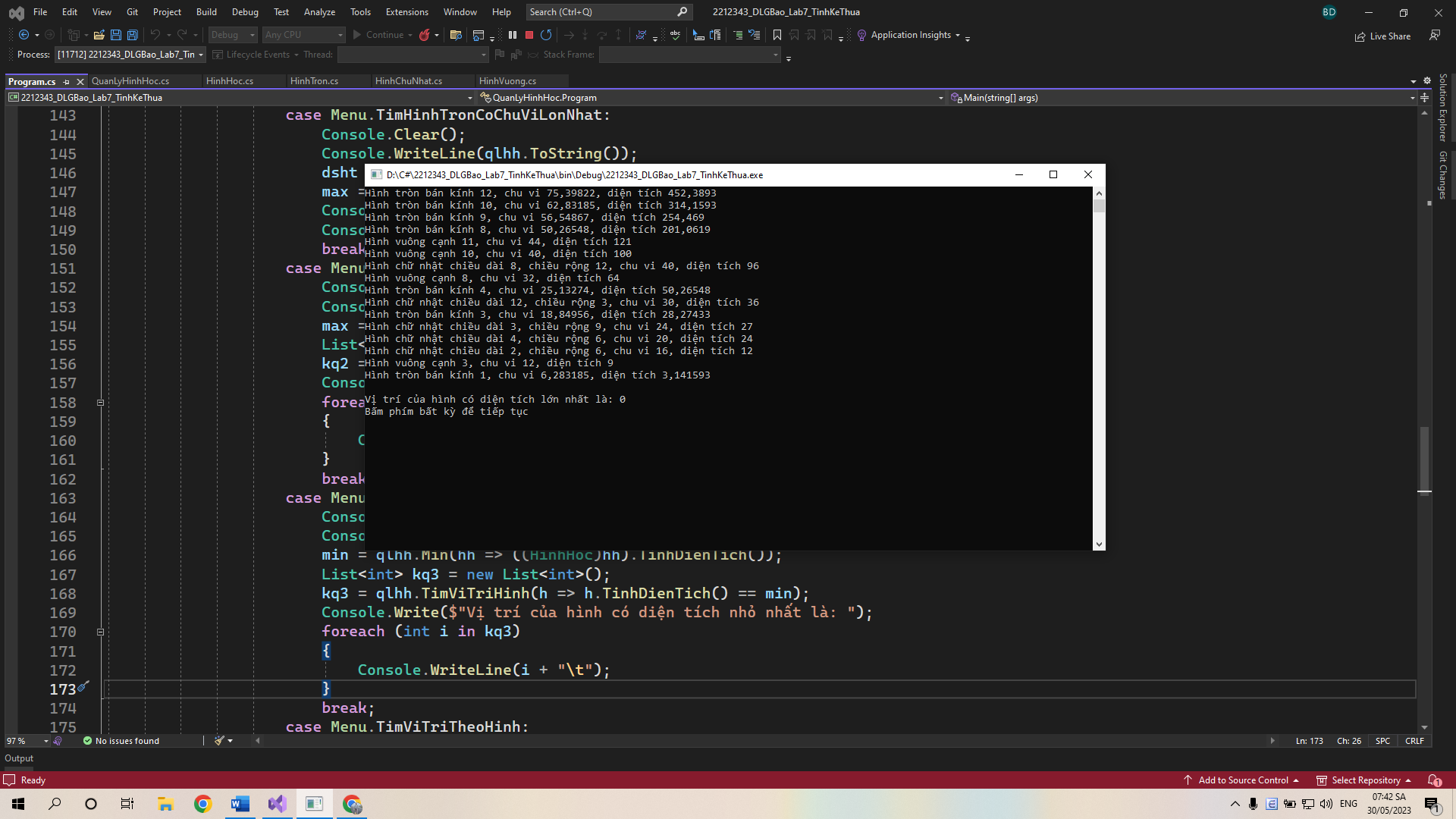
foreach (int i in kq3)

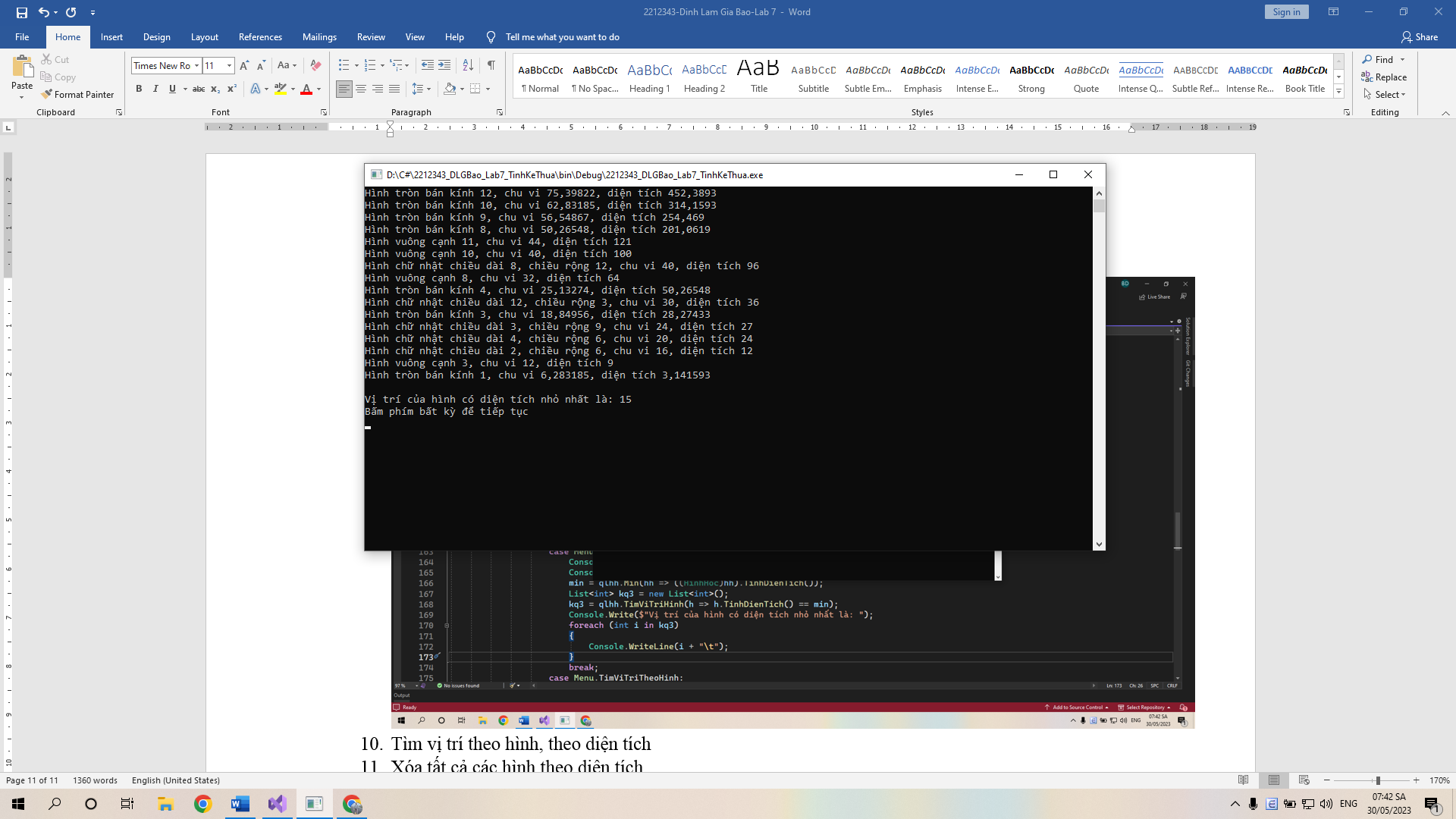
{

Console.WriteLine(i + "\t");

}

break;





1. Tìm vị trí theo hình, theo diện tích

case Menu.TimViTriTheoHinh:

Console.Clear();

Console.Write("Nhập hình: ");

string NhapLoaiHinh = Console.ReadLine();

List<int> kq = new List<int>();

switch (NhapLoaiHinh)

{

case "HV":

kq = qlhh.TimViTriHinh(h => h is HinhVuong);

break;

case "HCN":

kq = qlhh.TimViTriHinh(h => h is HinhChuNhat);

break;

case "HT":

kq = qlhh.TimViTriHinh(h => h is HinhTron);

break;

}

Console.WriteLine(qlhh.ToString());

Console.WriteLine($"Vị trí của {NhapLoaiHinh} là: ");

foreach (int i in kq)

{

Console.Write(i + " ");

}

break;

case Menu.TimViTriTheoDienTich:

Console.Clear();

Console.WriteLine(qlhh.ToString());

Console.Write("Nhập diện tích của hình: ");

List<int> kq1 = new List<int>();

float dt = float.Parse(Console.ReadLine());

kq1 = qlhh.TimViTriHinh(h => h.TinhDienTich() == dt);

Console.Write($"Vị trí của hình theo diện tích là: ");

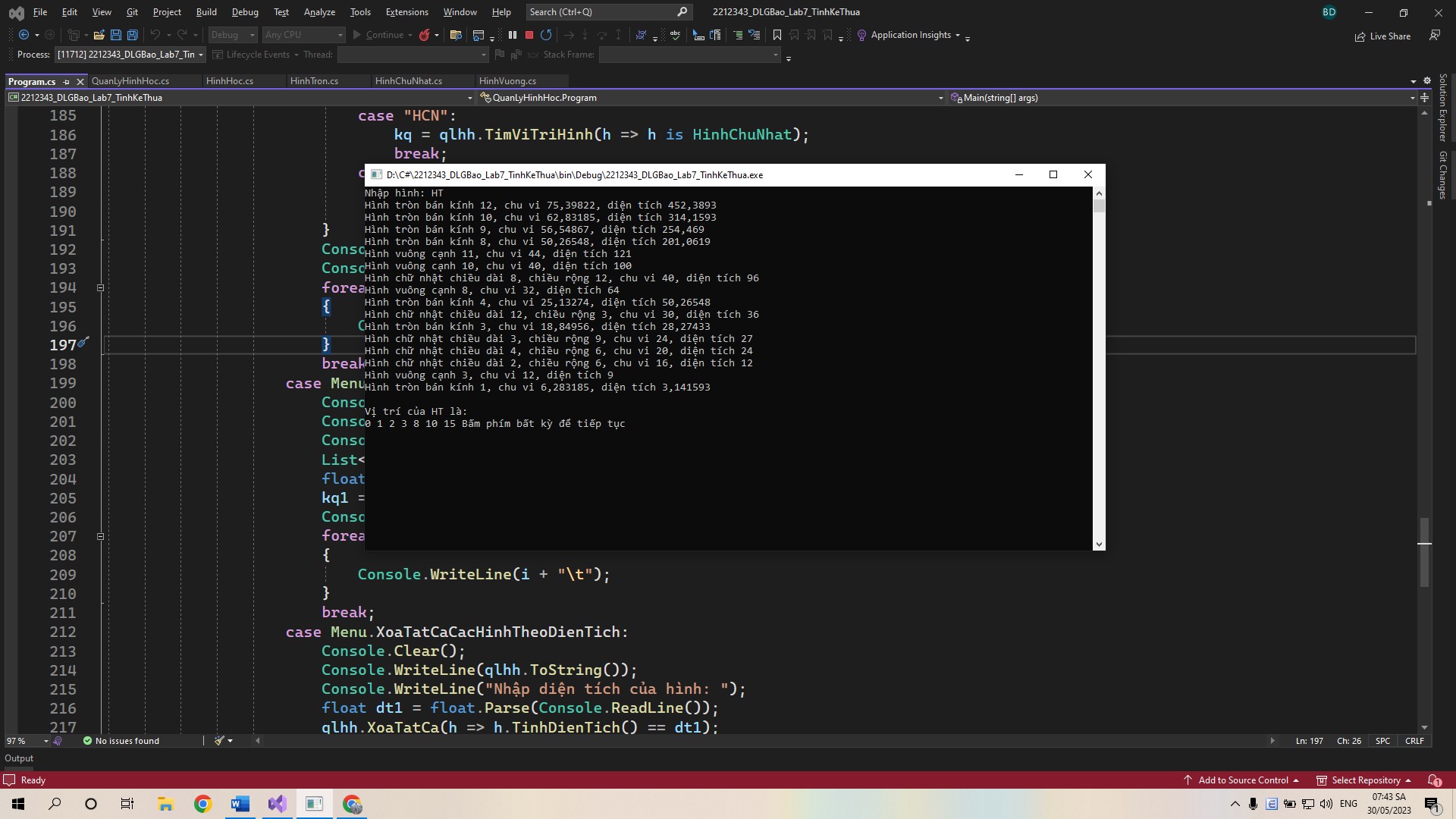
foreach (int i in kq1)

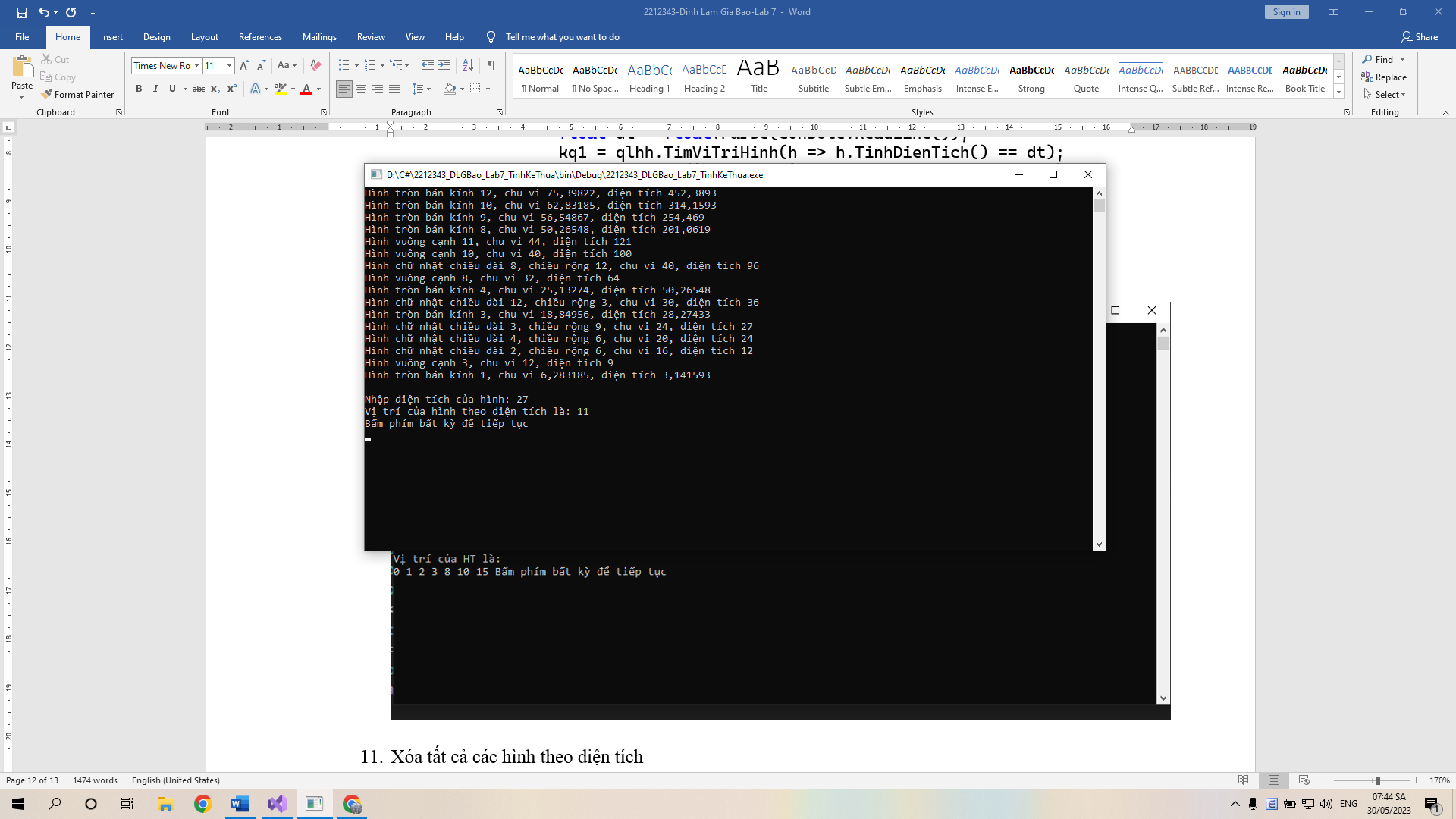
{

Console.WriteLine(i + "\t");

}

break;





1. Xóa tất cả các hình theo diện tích

case Menu.XoaTatCaCacHinhTheoDienTich:

Console.Clear();

Console.WriteLine(qlhh.ToString());

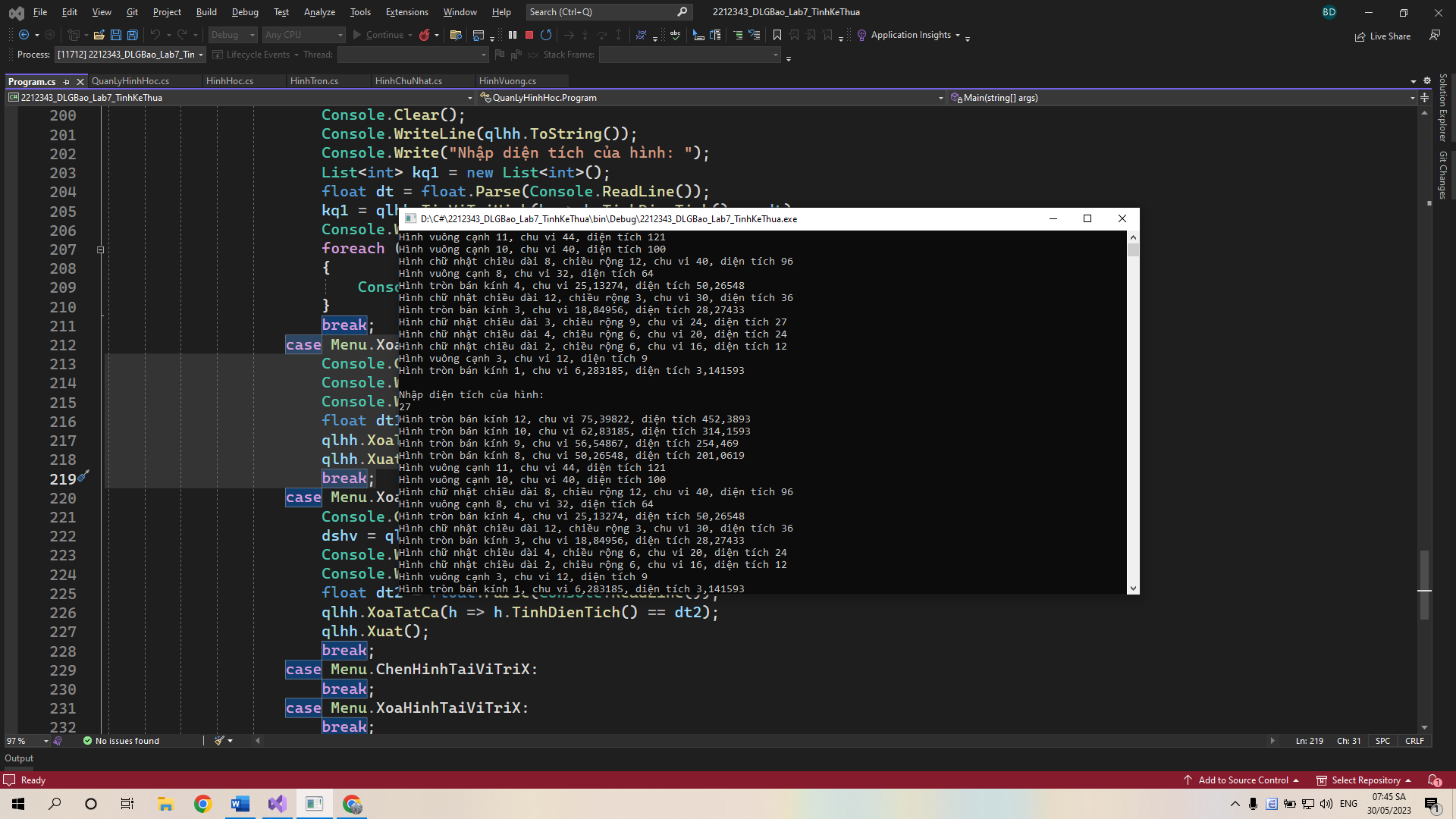
Console.WriteLine("Nhập diện tích của hình: ");

float dt1 = float.Parse(Console.ReadLine());

qlhh.XoaTatCa(h => h.TinhDienTich() == dt1);

qlhh.Xuat();

break;



1. Xóa hình vuông có diện tích là x

case Menu.XoaHinhVuongCoDienTichLaX:

Console.Clear();

dshv = qlhh.TimHinh(hh => hh is HinhVuong);

Console.WriteLine(dshv.ToString());

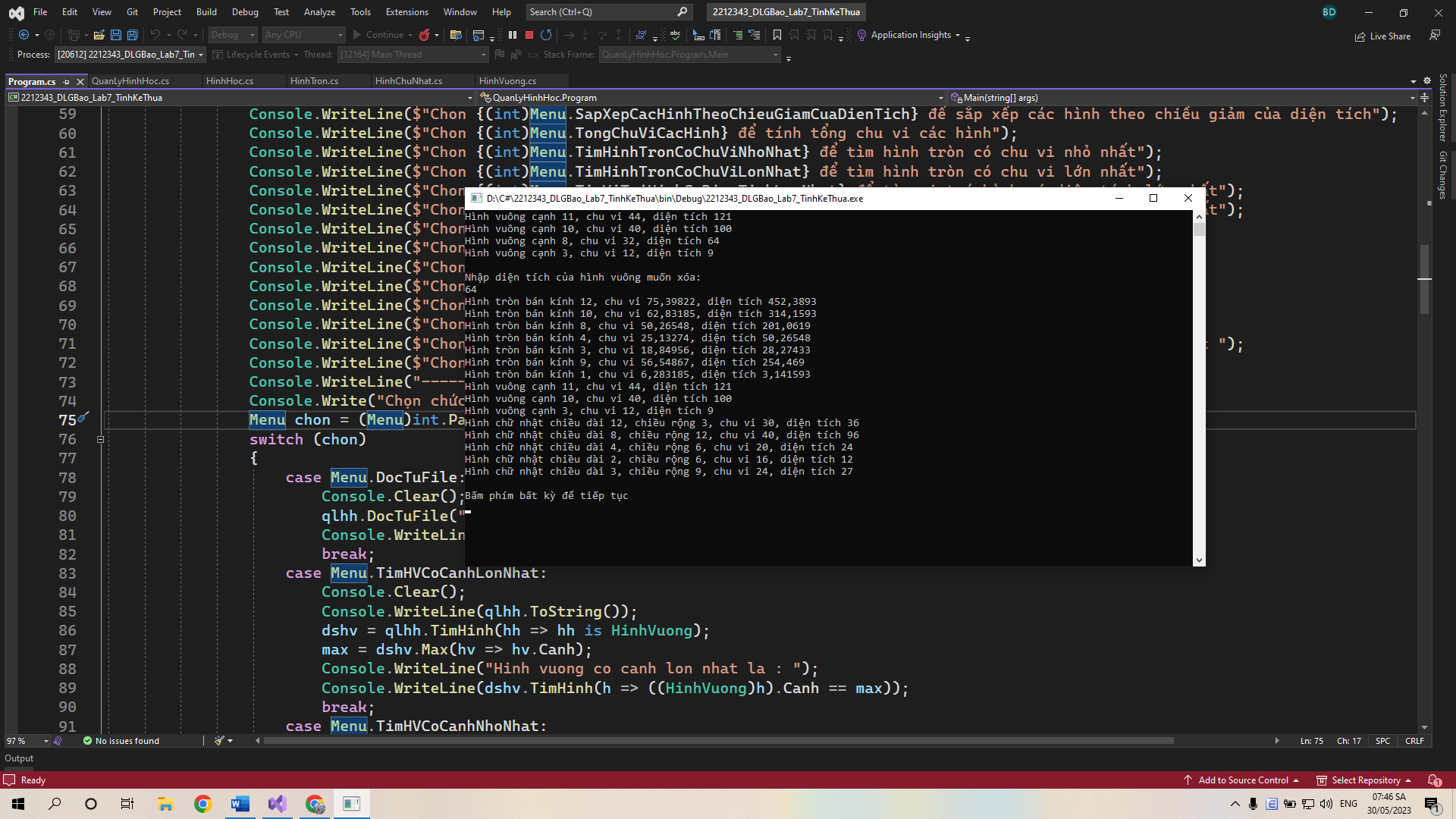
Console.WriteLine("Nhập diện tích của hình vuông muốn xóa: ");

float dt2 = float.Parse(Console.ReadLine());

qlhh.XoaTatCa(h => h.TinhDienTich() == dt2);

qlhh.Xuat();

break;



1. Chèn hình tại vị trí x
2. Xóa hình tại vị trí x
3. Tìm loại hình có số lượng nhiều nhất, ít nhất
4. Sử dụng enum tạo thực đơn cho các yêu câu trên

public enum Menu

{

DocTuFile = 1,

TimHVCoCanhLonNhat,

TimHVCoCanhNhoNhat,

TimHCNCoChieuDaiNganNhat,

TimHCNCoChieuDaiDaiNhat,

SapXepCacHinhTheoChieuTangCuaDienTich,

SapXepCacHinhTheoChieuGiamCuaDienTich,

TongChuViCacHinh,

TimHinhTronCoChuViNhoNhat,

TimHinhTronCoChuViLonNhat,

TimViTriHinhCoDienTichLonNhat,

TimViTriHinhCoDienTichNhoNhat,

TimViTriTheoHinh,

TimViTriTheoDienTich,

XoaTatCaCacHinhTheoDienTich,

XoaHinhVuongCoDienTichLaX,

ChenHinhTaiViTriX,

XoaHinhTaiViTriX,

TimLoaiHinhCoSoLuongNhieuNhat,

TimLoaiHinhCoSoLuongItNhat

}