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:
* Dòng xuất đầu thì sẽ xuất ra như bình thường
* dòng xuất thứ bởi vì đối tượng được lấy từ lớp con nên nó sẽ xuất ra "Hello" như khởi tạo
* c=cha; không được do lớp cha không có các thuộc tính và hàm từ lớp con.
* dòng xuất 3,4 nó xuất ra đúng như đối tượng khởi tạo vì các methods ở đúng class của nó
* trong vòng lặp
* trừ các đối tượng cha ra các đối tượng con được xuất ra 2 lần bởi vì lớp con được kế thừa từ lớp cha nên sẽ dùng được method XuatCha

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:

* Bổ sung lớp hình chữ nhật
* Nhập từ File
* Tìm hình vuông có cạnh lớn nhất, nhỏ nhất
* Tìm hình chữ nhật có chiều dài ngắn nhất, dài nhất
* Sắp xếp các hình theo chiều tăng giảm của diện tích
* Tính tổng chu vi của các hình
* Tìm hình tròn có chu vi nhỏ nhất, lớn nhất
* Ghi vào File theo hình (Ví dụ: file HinhTron.txt thì ghi hình tròn).
* Tìm vị trí của hình có diện tích lớn nhất, nhỏ nhất
* Tìm vị trí theo hình, theo diện tích
* Xóa tất cả các hình theo diện tích
* Xóa hình vuông có diện tích là x
* Chèn hình tại vị trí x
* Xóa hình tại vị trí x
* Tìm loại hình có số lượng nhiều nhất, ít nhất
* Sử dụng enum tạo thực đơn cho các yêu câu trên

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

public float TinhTongDT()

{

float tong = 0;

foreach (var item in collection)

{

tong += item.TinhDT();

}

return tong;

}

public void Xuat()

{

foreach (var item in collection)

{

item.Xuat();

}

}

public void NhapTuFile()

{

string tenFile = "test.csv";

StreamReader s = new StreamReader(tenFile);

string a = "";

while ((a = s.ReadLine()) != null)

{

if (a.Contains("hinhtron") && a != null)

{

HinhTron n = new HinhTron(a);

collection.Add(n);

}

if (a.Contains("hinhvuong") && a != null)

{

HinhVuong n = new HinhVuong(a);

collection.Add(n);

}

if (a.Contains("hinhchunhat") && a != null)

{

HinhChuNhat n = new HinhChuNhat(a);

collection.Add(n);

}

}

}

public List<HinhVuong> TimCanhNhoNhat()

{

List<HinhHoc> a = collection;

List<HinhVuong> b = a.OfType<HinhVuong>().ToList();

float minCanh = b.Min(HinhVuong => HinhVuong.Canh);

List<HinhVuong> c = b.Where(HinhVuong => HinhVuong.Canh == minCanh).ToList();

return c;

}

public List<HinhVuong> TimCanhLonNhat()

{

List<HinhVuong> b = collection.OfType<HinhVuong>().ToList();

float maxCanh = b.Max(HinhVuong => HinhVuong.Canh);

List<HinhVuong> c = b.Where(HinhVuong => HinhVuong.Canh == maxCanh).ToList();

return c;

}

public List<HinhChuNhat> TimCanhDaiNhoNhat()

{

List<HinhChuNhat> a = collection.OfType<HinhChuNhat>().ToList();

float minDai = a.Min(HinhChuNhat => HinhChuNhat.chieuDai);

List<HinhChuNhat> b = a.Where(HinhChuNhat => HinhChuNhat.chieuDai == minDai).ToList();

return b;

}

public List<HinhChuNhat> TimCanhDaiLonNhat()

{

List<HinhChuNhat> a = collection.OfType<HinhChuNhat>().ToList();

float maxDai = a.Max(HinhChuNhat => HinhChuNhat.chieuDai);

List<HinhChuNhat> b = a.Where(HinhChuNhat => HinhChuNhat.chieuDai == maxDai).ToList();

return b;

}

public List<HinhHoc> SapDienTichTheoChieuTang()

{

List<HinhHoc> a = collection;

var b = a.OrderBy(HinhHoc => HinhHoc.TinhDT()).ToList();

return b;

}

public List<HinhHoc> SapDienTichTheoChieuGiam()

{

List<HinhHoc> a = collection;

var b = a.OrderByDescending(HinhHoc => HinhHoc.TinhDT()).ToList();

return b;

}

public float TinhTongCV()

{

float tong = 0;

foreach (var h in collection)

{

tong += h.TinhChuVi();

}

return tong;

}

public List<HinhTron> TimHinhTronCoChuViMin()

{

List<HinhTron> a = collection.OfType<HinhTron>().ToList();

float minCanh = a.Min(HinhTron => HinhTron.TinhChuVi());

List<HinhTron> b = a.Where(HinhTron => HinhTron.TinhChuVi() == minCanh).ToList();

return b;

}

public List<HinhTron> TimHinhTronCoChuViMax()

{

List<HinhTron> a = collection.OfType<HinhTron>().ToList();

float maxCanh = a.Max(HinhTron => HinhTron.TinhChuVi());

List<HinhTron> b = a.Where(HinhTron => HinhTron.TinhChuVi() == maxCanh).ToList();

return b;

}

public List<HinhHoc> TimDienTichMin()

{

List<HinhHoc> a = collection;

float minCanh = a.Min(HinhHoc => HinhHoc.TinhDT());

List<HinhHoc> b = a.Where(HinhHoc => HinhHoc.TinhDT() == minCanh).ToList();

return b;

}

public List<HinhHoc> TimDienTichMax()

{

List<HinhHoc> a = collection;

float maxCanh = a.Max(HinhHoc => HinhHoc.TinhDT());

List<HinhHoc> b = a.Where(HinhHoc => HinhHoc.TinhDT() == maxCanh).ToList();

return b;

}

public List<HinhHoc> TimViTri(float a, string h)

{

List<HinhHoc> b = collection.Where(x => x.type == h && x.TinhDT() == a).ToList();

return b;

}

public List<HinhHoc> XoaHinhTheoDienTich(float b)

{

List<HinhHoc> a = collection;

foreach (var item in a)

{

if (item.TinhDT() == b)

{

a.Remove(item);

}

}

return a;

}

public List<HinhVuong> XoaHinhVuongTheoX(float x)

{

List<HinhVuong> a = collection.OfType<HinhVuong>().ToList();

foreach (var item in a)

{

if (item.TinhDT() == x)

{

a.Remove(item);

}

}

return a;

}

public void ChenHinh(int x)

{

Console.WriteLine("Chọn : HinhVuong, HinhTron, HinhChuNhat");

string a = Console.ReadLine();

if (a == "HinhVuong")

{

Console.WriteLine("Nhập cạnh");

float b = (float)int.Parse(Console.ReadLine());

HinhVuong h = new HinhVuong(b);

collection.Insert(x, h);

}

if (a == "HinhTron")

{

Console.WriteLine("Nhập bán kính");

float b = (float)int.Parse(Console.ReadLine());

HinhTron h = new HinhTron(b);

collection.Insert(x, h);

}

if (a == "HinhChuNhat")

{

Console.WriteLine("Nhập chiều dài, chiều rộng");

float b = (float)int.Parse(Console.ReadLine());

float c = (float)int.Parse(Console.ReadLine());

HinhChuNhat h = new HinhChuNhat(b, c);

collection.Insert(x, h);

}

}

public void XoaHinh(int x)

{

collection.RemoveAt(x);

}

private int Dem(string vl)

{

int dem = 0;

foreach (var item in collection)

{

if (item.type == vl)

dem++;

}

return dem;

}

public List<string> LoaiHinhNhieuNhat()

{

List<string> a = new List<string>();

int max = int.MinValue;

foreach (var item in collection)

{

if (max < Dem(item.type))

max = Dem(item.type);

}

foreach(var b in collection)

{

if (Dem(b.type) == max)

a.Add(b.type);

}

return a;

}

public List<string> LoaiHinhItNhat()

{

List<string> a = new List<string>();

int min = int.MaxValue;

foreach (var item in collection)

{

if (min > Dem(item.type))

min = Dem(item.type);

}

foreach(var b in collection)

{

if (Dem(b.type) == min)

a.Add(b.type);

}

return a;

}