using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Vector

{

    internal class TSquare

    {

        protected double a;

        public TSquare(double a)

        {

            this.a = a;

        }

        public TSquare()

        {

            a = 1;

        }

        public TSquare(TSquare square)

        {

            a = square.a;

        }

        public virtual void SetValue(double a)

        {

            if( a > 0)

                this.a = a;

        }

        public override string ToString()

        {

            return $"[a = {a}]";

        }

        public virtual double Square() => a \* a;

        public virtual double Perimeter() => 4 \* a;

        public bool IsEqulas(TSquare square) => a == square.a;

        public static TSquare operator +(TSquare square1, TSquare square2)=> new TSquare(square1.a + square2.a);

        public static TSquare operator -(TSquare square1, TSquare square2) => new TSquare(square1.a - square2.a);

        public static TSquare operator \*(double n, TSquare square2) => new TSquare(n \* square2.a);

        public static TSquare operator \*(TSquare square1, double n) => new TSquare(square1.a \*n);

    }

}

namespace Vector

{

    internal class TCube : TSquare

    {

        public TCube()

        {

            a = 1;

        }

        public TCube(double a):base(a)

        {

        }

        public TCube(TCube cube)

        {

            a=cube.a;

        }

        public double Volume() => a \* a \* a;

        public override double Square()

        {

            return base.Square() \* 6;

        }

        public override double Perimeter()

        {

            return base.Perimeter() \* 3;

        }

    }

}