==========

**Міністерство освіти і науки України**
**Національний технічний університет**
**«Дніпровська політехніка»**


# ЗВІТ
## про виконання лабораторних робіт
## з дисципліни
## «Об'єктно-орієнтоване програмування»
**Лабораторна робота № 2**

Виконав:
студент гр. 121-19-2
Назаркин С. А.

Прийняв:
викладач каф.
Приходченко С.Д.

**Дніпро**
**2020**

Варіант 17

Фігура: паралелограм.

Код класу:

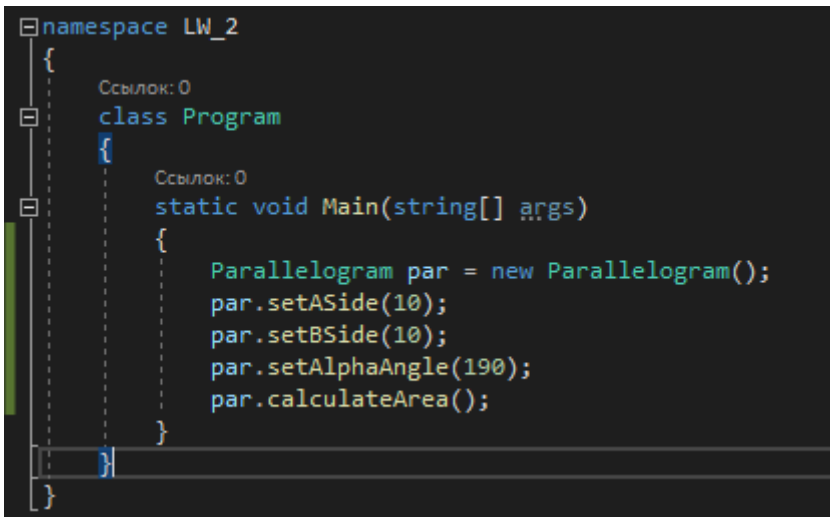
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;

namespace LW_2
{
    class Parallelogram
    {
        private string aSide;
        private string bSide;
        private string alphaAngle;
        private string betaAngle;
        private string aHeight;
        private string bHeight;
        public Parallelogram()
        {
            Console.Write("Call some setter methods to set values.\n");
            aSide = "0";
            bSide = "0";
            alphaAngle = "0";
            betaAngle = "0";
            aHeight = "0";
            bHeight = "0";
        }

        public void setASide(float newASideValue)
        {
            if (Convert.ToSingle(bSide) != newASideValue & newASideValue > 0)
            {
                aSide = Convert.ToString(newASideValue);
            }
            else
            {
                Console.Write("The aSide value isn't correct. Set a valid value.\n");
            }

        }

        public float getASide()
        {
            return Convert.ToSingle(aSide);
        }

        public void setBSide(float newBSideValue)
        {
            if (Convert.ToSingle(aSide) != newBSideValue & newBSideValue > 0)
            {
                bSide = Convert.ToString(newBSideValue);
            }
            else
            {
                Console.Write("The bSide value isn't correct. Set a valid value.\n");
            }
        }

        public float getBSide()
        {
            return Convert.ToSingle(bSide);
```

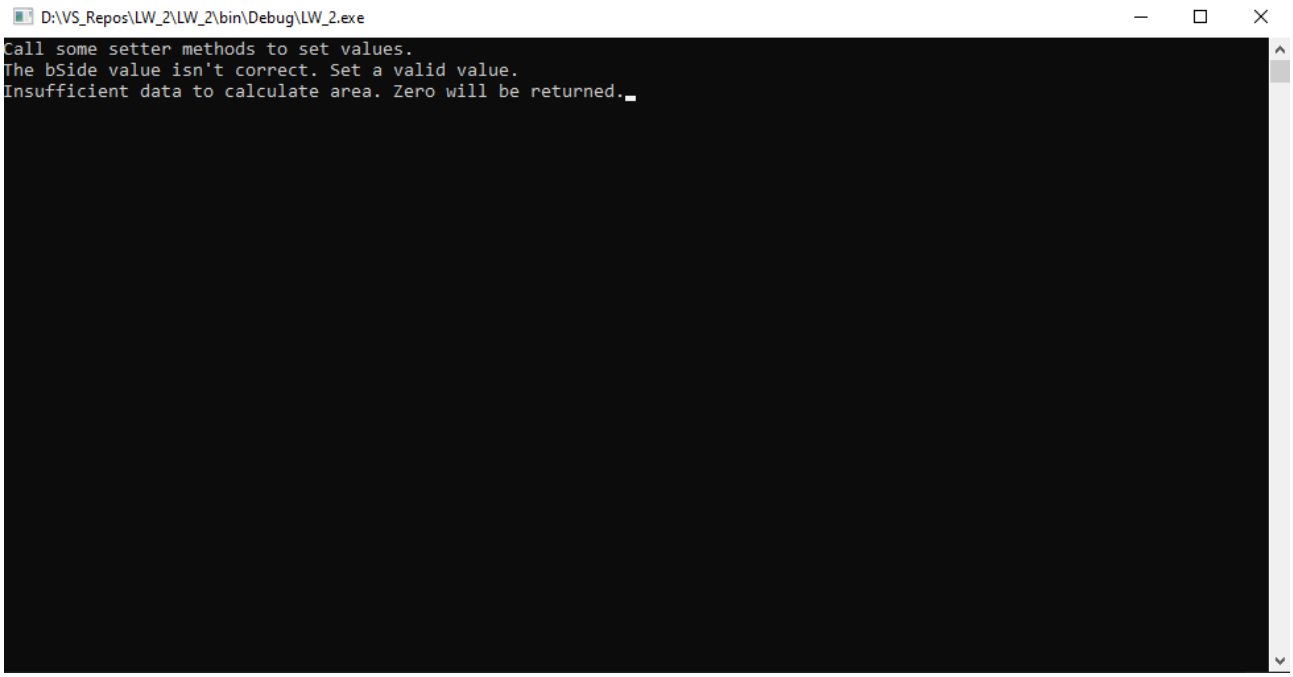
```
        }

        public void setAlphaAngle(float newAlphaAngleValue)
        {
            if (betaAngle == "0" & newAlphaAngleValue > 0 & newAlphaAngleValue != 90 &
newAlphaAngleValue < 180)
            {
                alphaAngle = Convert.ToString(newAlphaAngleValue);
            }
            if (betaAngle != "0")
            {
                alphaAngle = Convert.ToString(180 - Convert.ToSingle(betaAngle));
                Console.Write("The only possible alpha angle value is " + alphaAngle+ "\n");
            }
        }

        public float getAlphaAngle()
        {
            return Convert.ToSingle(alphaAngle);
        }

        public void setBetaAngle(float newBetaAngleValue)
        {
            if (alphaAngle == "0" & newBetaAngleValue > 0 & newBetaAngleValue != 90 &
newBetaAngleValue < 180)
            {
                betaAngle = Convert.ToString(newBetaAngleValue);
            }
            if (alphaAngle != "0")
            {
                betaAngle = Convert.ToString(180 - Convert.ToSingle(alphaAngle));
                Console.Write("The only possible beta angle value is " + betaAngle+ "\n");
            }
        }

        public float getBetaAngle()
        {
            return Convert.ToSingle(betaAngle);
        }

        public void setAHeight(float newHeightValue)
        {
            if (aSide != "0" & alphaAngle != "0")
            {
                aHeight = Convert.ToString(Convert.ToSingle(aSide) *
Math.Sin((Convert.ToInt32(alphaAngle) / 180D) * Math.PI));
                Console.Write("aSide * sind(alphaAngle) = " + aHeight + "\n");
            }
            else
            {
                if(newHeightValue > 0)
                {
                    aHeight = Convert.ToString(newHeightValue);
                }
                else
                {
                    Console.Write("The aHeight value isn't correct. Set a valid value.\n");
                }
            }
        }

        public float getAHeight()
        {
            return Convert.ToSingle(aHeight);
        }
```

```
        public void setBHeight(float newHeightValue)
        {
            if (bSide != "0" & betaAngle != "0")
            {
                bHeight = Convert.ToString(Convert.ToSingle(bSide) *
Math.Sin((Convert.ToInt32(betaAngle) / 180D) * Math.PI));
                Console.Write("aSide * sind(betaAngle) = " + bHeight + "\n");
            }
            else
            {
                if (newHeightValue > 0)
                {
                    bHeight = Convert.ToString(newHeightValue);
                }
                else
                {
                    Console.Write("The bHeight value isn't correct. Set a valid value.\n");
                }
            }
        }

        public float getBHeight()
        {
            return Convert.ToSingle(bHeight);
        }

        public float calculateArea()
        {
            float area = 0;
            if (aSide != "0" & aHeight != "0")
            {
                area = Convert.ToSingle(aSide) * Convert.ToSingle(aHeight);
                Console.Write("aSide * aHeight = "+ area);
                Console.ReadKey();
                return area;
            }
            else{
                if (bSide != "0" & bHeight != "0"){
                    area = Convert.ToSingle(bSide) * Convert.ToSingle(bHeight);
                    Console.Write("bSide * bHeight = " + area);
                    Console.ReadKey();
                    return area;
                }
                else{
                    if (aSide != "0" & bSide != "0" & alphaAngle != "0")
                {
                        area = Convert.ToSingle(aSide) * Convert.ToSingle(bSide) *
Convert.ToSingle(Math.Sin((Convert.ToInt32(betaAngle) / 180D) * Math.PI));
                        Console.Write("aSide * bSide * sin(alphaAngle) = " + area);
                        Console.ReadKey();
                        return area;
                    }
                    else{
                        if (aSide != "0" & bSide != "0" & betaAngle != "0")
                        {
                            area = Convert.ToSingle(aSide) * Convert.ToSingle(bSide) *
Convert.ToSingle(Math.Sin((Convert.ToInt32(betaAngle) / 180D) * Math.PI));
                            Console.Write("aSide * bSide * sin(betaAngle) = " + area);
                            Console.ReadKey();
                            return area;
                        }
                        else{
                            Console.Write("Insufficient data to calculate area. Zero will be
returned.");
```

```
                        Console.ReadKey();
                        return area;
                    }
                }
            }
        }
    }

    ~Parallelogram() { }
  }
}
```

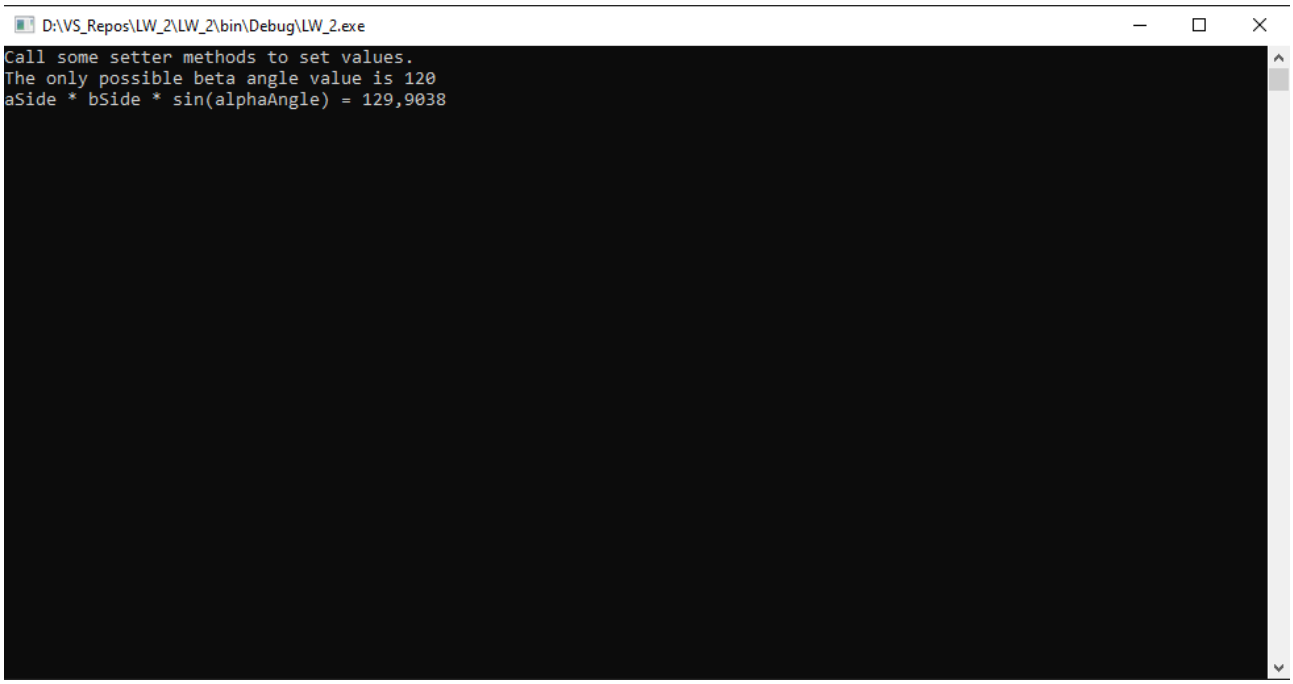
Результати:



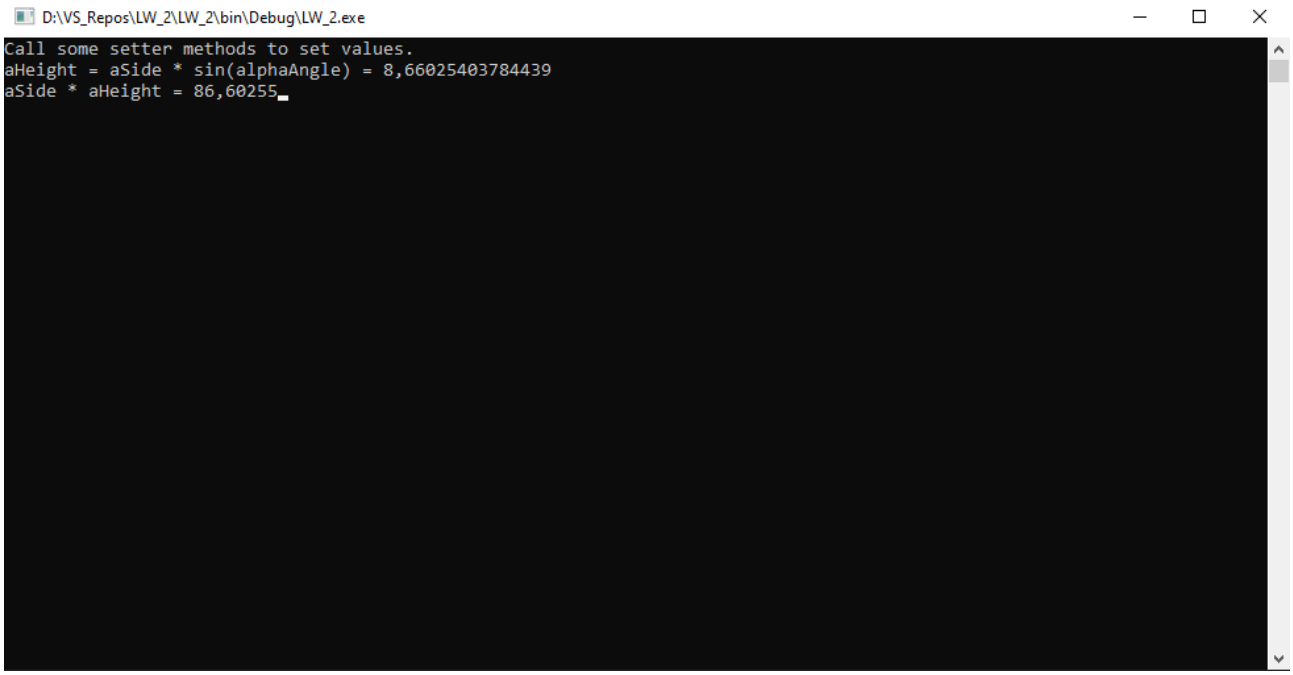

В паралелограмі сторони дорівнюють попарно, тому aSide не може дорівнювати bSide. Також кут не може дорівнювати 180 та більше. В даному випадку площина не розраховується, тому що кут та сторона не були присвоєні.

```
namespace LW_2
{
    Ссылок: 0
    class Program
    {
        Ссылок: 0
        static void Main(string[] args)
        {
            Parallelogram par = new Parallelogram();
            par.setAlphaAngle(60);
            par.setBetaAngle(140);
            par.setASide(10);
            par.setBSide(15);
            par.calculateArea();
        }
    }
}
```




В даному випадку величина сторін та кута альфа допустима, а величина бета — ні. Сума цих кутів повинна дорівнювати 180. Тому значення 140 не було присвоєне, проте було присвоєне 120. Для розрахунку площини достатньо 2 сторони та кут.

```
namespace LW_2
{
    Ссылок: 0
    class Program
    {
        Ссылок: 0
        static void Main(string[] args)
        {
            Parallelogram par = new Parallelogram();
            par.setASide(10);
            par.setAlphaAngle(60);
            par.setAHeight(30);
            par.calculateArea();
        }
    }
}
```




Так як була задана сторона а та кут альфа, ми можемо розрахувати висоту а. Задана висота неправильна. Тому була присвоєна розрахована висота. Та знайдена площина через сторону та висоту.