**ЛАБОРАТОРНА РОБОТА № 7**

**ООП: наслідування класів, віртуальні методи.**

***Мета:*** засвоїти принципи реалізації класів за допомогою наслідування.

**1. Створити рішення з назвою «oop-lab7».**

Рішення має складатися з двох частин – бібліотеки класів і консольного проекту. У бібліотеці класів реалізувати вказані у варіанті класи з використанням наслідування, а у консольному проекті – реалізувати створення об’єктів та виклик методів.

**2. Реалізувати такі класи, використавши наслідування:**

- «Людина» (ім’я, прізвище, дата народження);

- «Абітурієнт» (кількість балів сертифікатів ЗНО, кількість балів за документ про освіту, назва загальноосвітнього навчального закладу);

- «Студент» (курс, група, факультет, вищий навчальний заклад);

- «Викладач» (посада, кафедра, вищий навчальний заклад);

- «Користувач бібліотеки» (номер читацького квитка, дата видачі, розмір щомісячного читацького внеску);

Для кожного класу передбачити:

- конструктор по замовчуванню;

- два конструктори з параметрами;

- конструктор копіювання;

- методи для встановлення та читання значень;

- передбачити віртуальний метод ShowInfo(), який виводить усю доступну інфор-мацію (включаючи поля батьківських класів).

Написати код, у якому створюються об’єкти усіх створених класів та реалізувати ви-клик реалізованих методів.

Лістинг програми (методу main):

using System;

using System.Text;

using static System.Console;

using ClassLibrary;

namespace ConsoleExs1

{

class Program

{

static void Main(string[] args)

{

System.Globalization.CultureInfo customCulture = (System.Globalization.CultureInfo)

System.Threading.Thread.CurrentThread.CurrentCulture.Clone();

customCulture.NumberFormat.NumberDecimalSeparator = ".";

System.Threading.Thread.CurrentThread.CurrentCulture = customCulture;

OutputEncoding = Encoding.Unicode;

InputEncoding = Encoding.Unicode;

Title = "Лабораторна робота №7";

SetWindowSize(130, 25);

Person person = new Person();

Student student = new Student("Олександр", "Сівченко", new DateTime(2001, 1, 5), 1, "ПІ-60", "ФІКТ", "ЖДТУ");

LibraryUser libraryUser = new LibraryUser("Дмитро", "Воронков", new DateTime(2000, 9, 21), 4728, new DateTime(2019, 8, 10), 10);

Person[] people = {

new Person(),

new Entrant(),

new Student(student),

new Teacher(),

new LibraryUser(libraryUser),

null

};

person.PrintPersonTableInConsole();

Write("\n");

foreach (var i in people)

{

if (i != null)

{

i.ShowInfo();

}

}

}

}

}

Лістинг бібліотеки класів ***ClassLibrary***

**Клас Person**

using System;

using static System.Console;

namespace ClassLibrary

{

public class Person

{

protected string name;

protected string surname;

protected DateTime DateOfBirth;

public string Name { get; set; }

public string Surname { get; set; }

public Person() : this("Марійка", "Харипончук", new DateTime(2001, 6, 12))

{

}

public Person(string name, string surname, DateTime dateOfBirth)

{

Name = name;

Surname = surname;

DateOfBirth = dateOfBirth;

}

public Person(string name, string surname)

{

Name = name;

Surname = surname;

}

public Person(Person obj)

{

Name = obj.Name;

Surname = obj.Surname;

DateOfBirth = obj.DateOfBirth;

}

public void PrintPersonTableInConsole()

{

ForegroundColor = ConsoleColor.DarkMagenta;

Write("\n\n| Ім'я | Прізвище | Дата народження |");

ResetColor();

}

public virtual void ShowInfo()

{

Write($"{Name,15}{Surname,15}{DateOfBirth.ToString("d"),20}");

}

}

}

**Клас Entrant**

using System;

using static System.Console;

namespace ClassLibrary

{

public class Entrant : Person

{

protected ZNO\_Points PointsZNO; //(кількість балів сертифікатів ЗНО

protected float pointsCertificate; //кількість балів за документ про освіту

protected string schoolName; //назва загальноосвітнього навчального закладу

public string SchoolName { set; get; }

public float PointsCertificate

{

set

{

if (value > 0 && value < 13)

pointsCertificate = value;

else

{

BackgroundColor = ConsoleColor.Red;

ForegroundColor = ConsoleColor.White;

WriteLine("Некоректне значення оцінки атестата!!!");

ResetColor();

}

}

get

{

return pointsCertificate;

}

}

public Entrant() : this("Марійка", "Харипончук", new DateTime(2001, 6, 12), "Українська мова та літ-ра", "Математика", "Історія", " ", 189, 182, 182, 0, (float)10.5, "Олевська гімназія")

{

}

public Entrant(string name, string surname, DateTime dateOfBirth, string firstSbj, string secondSbj, string thirdSbj, string fourthSbj, int firstPoints, int secondPoints, int thirdPoints, int fourthPoints, float pointsCertificate, string schoolName) : base(name, surname, dateOfBirth)

{

PointsZNO = new ZNO\_Points(firstSbj, secondSbj, thirdSbj, fourthSbj, firstPoints, secondPoints, thirdPoints, fourthPoints);

PointsCertificate = pointsCertificate;

SchoolName = schoolName;

}

public Entrant(Entrant obj) : base(obj)

{

PointsZNO = new ZNO\_Points(obj.PointsZNO);

PointsCertificate = obj.PointsCertificate;

SchoolName = obj.SchoolName;

}

public void PrintEntrantTableInConsole()

{

PrintPersonTableInConsole();

ForegroundColor = ConsoleColor.DarkMagenta;

WriteLine(" Бали ЗНО | Атестат | ВНЗ |");

ResetColor();

}

public override void ShowInfo()

{

BackgroundColor = ConsoleColor.Magenta;

ForegroundColor = ConsoleColor.White;

WriteLine("\nЛюдина");

ResetColor();

PrintEntrantTableInConsole();

base.ShowInfo();

WriteLine($"{PointsCertificate,47}{SchoolName,22}");

PointsZNO.ShowInfo();

BackgroundColor = ConsoleColor.Magenta;

ForegroundColor = ConsoleColor.White;

WriteLine("Абітурієнт");

ResetColor();

}

}

}

**Клас ZNO\_Points**

using System;

using static System.Console;

namespace ClassLibrary

{

public class ZNO\_Points

{

protected int firstPoints;

protected int secondPoints;

protected int thirdPoints;

protected int fourthPoints;

public string FirstSubject { set; get; }

public string SecondSubject { set; get; }

public string ThirdSubject { set; get; }

public string FourthSubject { set; get; }

public int FirstPoints

{

set

{

if (value >= 0 && value <= 200)

firstPoints = value;

else

{

BackgroundColor = ConsoleColor.Red;

ForegroundColor = ConsoleColor.White;

WriteLine("Некоректне значення балів за перший предмет!!!");

ResetColor();

}

}

get

{

return firstPoints;

}

}

public int SecondPoints

{

set

{

if (value >= 0 && value <= 200)

secondPoints = value;

else

{

BackgroundColor = ConsoleColor.Red;

ForegroundColor = ConsoleColor.White;

WriteLine("Некоректне значення балів за другий предмет!!!");

ResetColor();

}

}

get

{

return secondPoints;

}

}

public int ThirdPoints

{

set

{

if (value >= 0 && value <= 200)

thirdPoints = value;

else

{

BackgroundColor = ConsoleColor.Red;

ForegroundColor = ConsoleColor.White;

WriteLine("Некоректне значення балів за третій предмет!!!");

ResetColor();

}

}

get

{

return thirdPoints;

}

}

public int FourthPoints

{

set

{

if (value >= 0 && value <= 200)

fourthPoints = value;

else

{

BackgroundColor = ConsoleColor.Red;

ForegroundColor = ConsoleColor.White;

WriteLine("Некоректне значення балів за четвертий предмет!!!");

ResetColor();

}

}

get

{

return fourthPoints;

}

}

public ZNO\_Points(string firstSbj, string secondSbj, string thirdSbj, int firstPoints, int secondPoints, int thirdPoints) : this(firstSbj, secondSbj, thirdSbj, " ", firstPoints, secondPoints, thirdPoints, 0)

{

}

public ZNO\_Points(string firstSbj, string secondSbj, string thirdSbj, string fourthSbj, int firstPoints, int secondPoints, int thirdPoints, int fourthPoints)

{

FirstSubject = firstSbj;

SecondSubject = secondSbj;

ThirdSubject = thirdSbj;

FourthSubject = fourthSbj;

FirstPoints = firstPoints;

SecondPoints = secondPoints;

ThirdPoints = thirdPoints;

FourthPoints = fourthPoints;

}

public ZNO\_Points(ZNO\_Points obj)

{

FirstSubject = obj.FirstSubject;

SecondSubject = obj.SecondSubject;

ThirdSubject = obj.ThirdSubject;

FourthSubject = obj.FourthSubject;

FirstPoints = obj.FirstPoints;

SecondPoints = obj.SecondPoints;

ThirdPoints = obj.ThirdPoints;

FourthPoints = obj.FourthPoints;

}

public void ShowInfo()

{

if (FirstSubject != " ")

WriteLine($"{FirstSubject,80}:{FirstPoints,4}");

if (SecondSubject != " ")

WriteLine($"{SecondSubject,80}:{SecondPoints,4}");

if (ThirdSubject != " ")

WriteLine($"{ThirdSubject,80}:{ThirdPoints,4}");

if (FourthSubject != " ")

WriteLine($"{FourthSubject,80}:{FourthPoints,4}");

}

}

}

**Клас Student**

using System;

using static System.Console;

namespace ClassLibrary

{

public class Student : Person

{

protected int course; //курс

protected int group; //група

protected int faculty; //факультет

protected int universityName; //вищий навчальний заклад

public string Group { set; get; }

public string Faculty { set; get; }

public string UniversityName { set; get; }

public int Course

{

set

{

if (value > 0)

course = value;

else

{

BackgroundColor = ConsoleColor.Red;

ForegroundColor = ConsoleColor.White;

WriteLine("Некоректне значення курса!!!");

ResetColor();

}

}

get

{

return course;

}

}

public Student() : this("Марійка", "Харипончук", new DateTime(2001, 6, 12), 1, "ПІ-60", "ФІКТ", "ЖДТУ")

{

}

public Student(string name, string surname, DateTime dateOfBirth, int course, string group, string faculty, string universityName) : base(name, surname, dateOfBirth)

{

Course = course;

Group = group;

Faculty = faculty;

UniversityName = universityName;

}

public Student(Student obj) : base(obj)

{

Course = obj.Course;

Group = obj.Group;

Faculty = obj.Faculty;

UniversityName = obj.UniversityName;

}

public void PrintStudentTableInConsole()

{

PrintPersonTableInConsole();

ForegroundColor = ConsoleColor.DarkMagenta;

WriteLine(" Курс | Група | Факультет | ВНЗ |");

ResetColor();

}

public override void ShowInfo()

{

PrintStudentTableInConsole();

base.ShowInfo();

WriteLine($"{Course,6}{Group,11}{Faculty,13}{UniversityName,15}");

BackgroundColor = ConsoleColor.Magenta;

ForegroundColor = ConsoleColor.White;

WriteLine("Студент");

ResetColor();

}

}

}

**Клас Teacher**

using System;

using static System.Console;

namespace ClassLibrary

{

public class Teacher : Person

{

protected string position; //посада

protected string chair; //кафедра

protected string universityName; //вищий навчальний заклад

public string Position { set; get; }

public string Chair { set; get; }

public string UniversityName { set; get; }

public Teacher() : this("Віталій", "Левківський", new DateTime(1990, 5, 7), "Вчитель", "ФІКТ", "ЖДТУ")

{

}

public Teacher(string name, string surname, DateTime dateOfBirth, string position, string chair, string universityName) : base(name, surname, dateOfBirth)

{

Position = position;

Chair = chair;

UniversityName = universityName;

}

public Teacher(Teacher obj) : base(obj)

{

Position = obj.Position;

Chair = obj.Chair;

UniversityName = obj.UniversityName;

}

public void PrintTeacherTableInConsole()

{

PrintPersonTableInConsole();

ForegroundColor = ConsoleColor.DarkMagenta;

WriteLine(" Посада | Кафедра | ВНЗ |");

ResetColor();

}

override public void ShowInfo()

{

PrintTeacherTableInConsole();

base.ShowInfo();

WriteLine($"{Position,11}{Chair,11}{UniversityName,16}");

BackgroundColor = ConsoleColor.Magenta;

ForegroundColor = ConsoleColor.White;

WriteLine("Викладач");

ResetColor();

}

}

}

**Клас LibraryUser**

using System;

using static System.Console;

namespace ClassLibrary

{

public class LibraryUser : Person

{

protected int numOfReadersTicket; //номер читацького квитка

protected DateTime DateOfIssue; //дата видачі

protected int payment; // розмір щомісячного читацького внеску

public int NumOfReadersTicket

{

set

{

if (value > 0)

numOfReadersTicket = value;

else

{

BackgroundColor = ConsoleColor.Red;

ForegroundColor = ConsoleColor.White;

WriteLine("Некоректне значення номера читацького квитка!!!");

ResetColor();

}

}

get

{

return numOfReadersTicket;

}

}

public int Payment

{

set

{

if (value > 0)

payment = value;

else

{

BackgroundColor = ConsoleColor.Red;

ForegroundColor = ConsoleColor.White;

WriteLine("Некоректне значення розміру щомісячного читацького внеску!!!");

ResetColor();

}

}

get

{

return payment;

}

}

public LibraryUser() : this("Марійка", "Харипончук", new DateTime(2001, 6, 12), 4728, new DateTime(2019, 8, 10), 10)

{

}

public LibraryUser(string name, string surname, DateTime dateOfBirth, int numOfReadersTicket, DateTime dateOfIssue, int payment) : base(name, surname, dateOfBirth)

{

NumOfReadersTicket = numOfReadersTicket;

DateOfIssue = dateOfIssue;

Payment = payment;

}

public LibraryUser(LibraryUser obj) : base(obj)

{

NumOfReadersTicket = obj.numOfReadersTicket;

DateOfIssue = obj.DateOfIssue;

Payment = obj.payment;

}

public void PrintLibraryUserTableInConsole()

{

PrintPersonTableInConsole();

ForegroundColor = ConsoleColor.DarkMagenta;

WriteLine(" Номер читацького квитка | Дата видачі | Щомісячний читацький внесок |");

ResetColor();

}

override public void ShowInfo()

{

PrintLibraryUserTableInConsole();

base.ShowInfo();

WriteLine($"{NumOfReadersTicket,23}{DateOfIssue.ToString("d"),21}{Payment,28}грн");

BackgroundColor = ConsoleColor.Magenta;

ForegroundColor = ConsoleColor.White;

WriteLine("Користувач бібліотеки");

ResetColor();

}

}

}

Результат виконання програми:

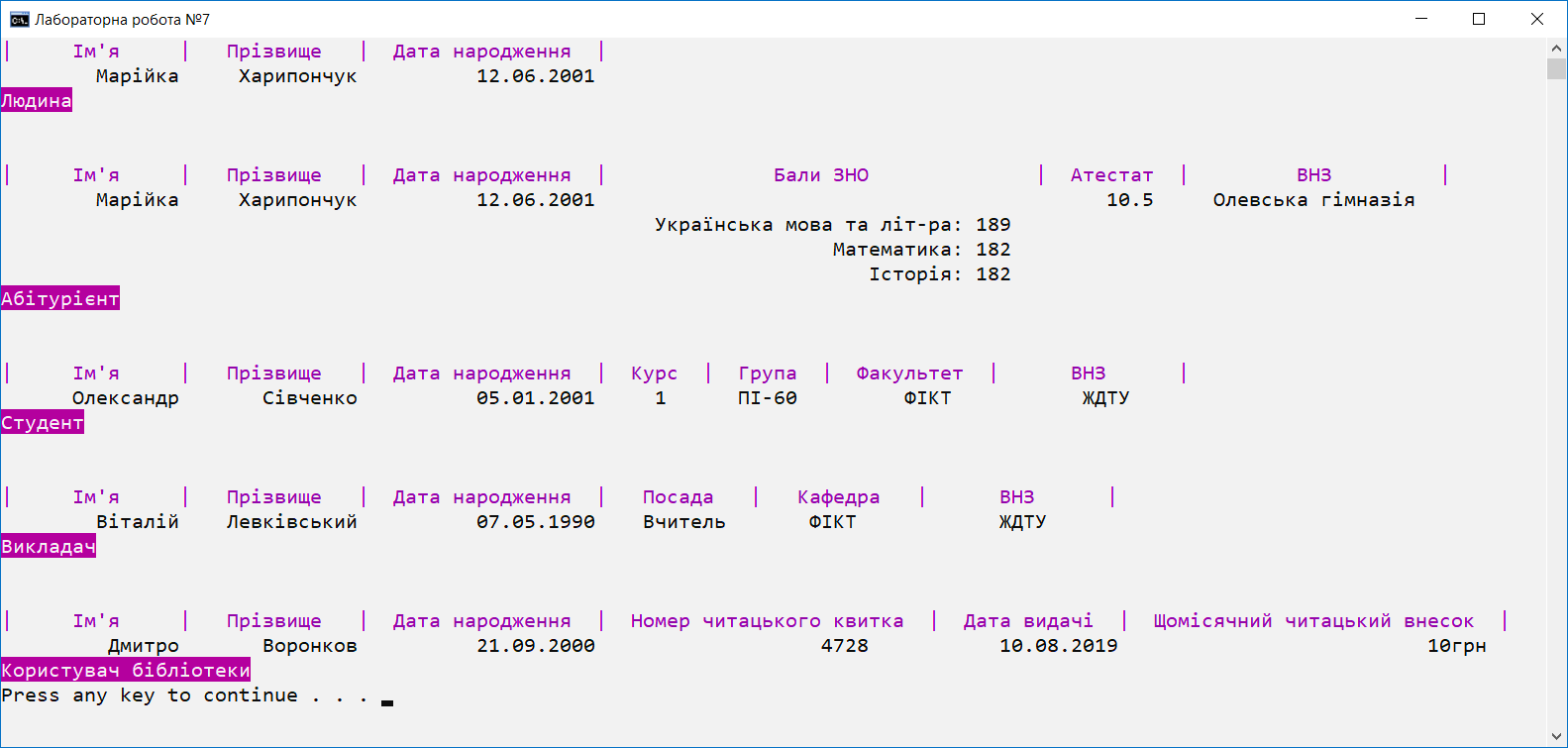
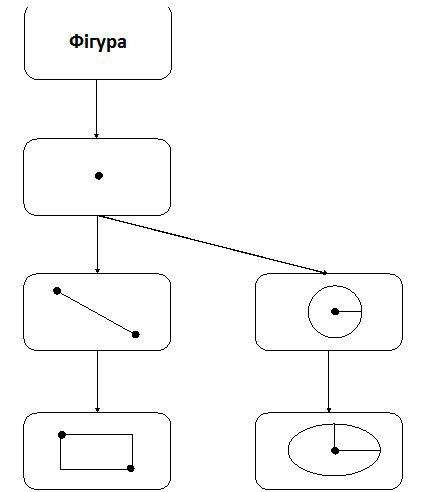


Рис. 1. Результат виконання роботи програми

**3. Створити віконний додаток:**

Реалізувати класи, показані на рисунку, використавши наслідування. Головним базо-вим класом для усіх об'єктів є клас Shape Point – фігура на площині з її координатами та кольором. Передбачте конструктор по-замовчуванню, конструктори з параметрами, конструк-тор копіювання, методи для зміни координат, розмірів та кольору. У конструкторах по замовчуванню кожного класу реалізуйте ініціалізацію полів випадковими даними. У кожному класі реалізувати метод Draw, який приймає об’єкт типу Graphics, на якому буде здійснюватися малювання фігури.Лістинг програми.



Лістинг програми (лістинг форми Windows):

using System;

using System.Drawing;

using System.Windows.Forms;

using ClassLibraryFigure;

namespace FormExs2

{

public partial class Form1 : Form

{

protected Bitmap BitCanvas;

protected Graphics canvas;

protected Random rand = new Random();

public Form1()

{

InitializeComponent();

BitCanvas = new Bitmap(pictureBoxCanvas.Width, pictureBoxCanvas.Height);

}

private void buttonDraw\_Click(object sender, EventArgs e)

{

Draws();

}

private void buttonErase\_Click(object sender, EventArgs e)

{

pictureBoxCanvas.Image = null;

}

private void Draws()

{

canvas = Graphics.FromImage(BitCanvas);

Shape[] array = new Shape[20];

for (int i = 0; i < array.Length; i++)

{

switch (rand.Next(6))

{

case 1:

array[i] = new Dot();

array[i].Draw(canvas);

break;

case 2:

array[i] = new Line();

array[i].Draw(canvas);

break;

case 3:

array[i] = new Circle(rand.Next(255), rand.Next(255), rand.Next(255), rand.Next(25), 5, 8, rand.Next(300), rand.Next(300));

array[i].Draw(canvas);

break;

case 4:

array[i] = new Ellipse(rand.Next(255), rand.Next(255), rand.Next(255), rand.Next(15), 5, 12, rand.Next(377), rand.Next(788));

array[i].Draw(canvas);

break;

case 5:

array[i] = new ClassLibraryFigure.Rectangle(rand.Next(300), rand.Next(300));

array[i].Draw(canvas);

break;

}

}

pictureBoxCanvas.Image = BitCanvas;

}

}

}

Лістинг бібліотеки класів ***ClassLibraryFigure***

**Клас Shape**

using System;

using System.Drawing;

namespace ClassLibraryFigure

{

abstract public class Shape

{

protected class ColorRGB

{

protected int red, green, blue;

public int Red

{

get

{

return red;

}

set

{

if (value >= 0 && value <= 255)

red = value;

}

}

public int Green

{

get

{

return green;

}

set

{

if (value >= 0 && value <= 255)

green = value;

}

}

public int Blue

{

get

{

return blue;

}

set

{

if (value >= 0 && value <= 255)

blue = value;

}

}

public ColorRGB()

{

Random rand = new Random();

Red = rand.Next(256);

Green = rand.Next(256);

Blue = rand.Next(256);

}

public ColorRGB(int red, int green, int blue)

{

Red = red;

Green = green;

Blue = blue;

}

public ColorRGB(ColorRGB obj)

{

Red = obj.Red;

Green = obj.Green;

Blue = obj.Blue;

}

}

protected class Point

{

protected int x, y;

public int X

{

set

{

if (value > 0)

x = value;

}

get

{

return x;

}

}

public int Y

{

set

{

if (value > 0)

y = value;

}

get

{

return y;

}

}

public Point()

{

Random rand = new Random();

X = rand.Next(200);

Y = rand.Next(200);

}

public Point(int maxY, int maxX)

{

Random rand = new Random();

X = rand.Next(maxX);

Y = rand.Next(maxY);

}

public Point(Point obj)

{

X = obj.X;

Y = obj.Y;

}

}

protected ColorRGB Color;

public int PenWidth { set; get; }

public abstract void Draw(Graphics canvas);

}

}

**Клас Dot**

using System;

using System.Drawing;

namespace ClassLibraryFigure

{

public class Dot : Shape

{

protected Point start;

public Dot()

{

Random rand = new Random();

Color = new ColorRGB();

PenWidth = rand.Next(10);

start = new Point();

}

public Dot(int maxY, int maxX)

{

Random rand = new Random();

Color = new ColorRGB();

PenWidth = rand.Next(10);

start = new Point(maxY, maxX);

}

public Dot(int red, int green, int blue, int penWidth, int startX, int startY)

{

Color = new ColorRGB(red, green, blue);

PenWidth = penWidth;

start = new Point(startX, startY);

}

public Dot(Dot obj)

{

Color = obj.Color;

PenWidth = obj.PenWidth;

start = obj.start;

}

public override void Draw(Graphics canvas)

{

Pen pen = new Pen(System.Drawing.Color.FromArgb(Color.Red, Color.Green, Color.Blue), PenWidth);

canvas.DrawRectangle(pen, start.X, start.Y, 1, 1);

}

}

}

**Клас Line**

using System;

using System.Drawing;

namespace ClassLibraryFigure

{

public class Line : Dot

{

protected Point end;

public Line() : base()

{

end = new Point();

}

public Line(int maxY, int maxX) : base(maxY, maxX)

{

end = new Point();

}

public Line(int red, int green, int blue, int penWidth, int startX, int startY, int endX, int endY)

: base(red, green, blue, penWidth, startX, startY)

{

end.X = endX;

end.Y = endY;

}

public Line(Line obj)

{

end = obj.end;

}

public override void Draw(Graphics canvas)

{

Pen pen = new Pen(System.Drawing.Color.FromArgb(Color.Red, Color.Green, Color.Blue), PenWidth);

canvas.DrawLine(pen, start.X, start.Y, end.X, end.Y);

}

}

}

**Клас Circle**

using System;

using System.Drawing;

namespace ClassLibraryFigure

{

public class Circle : Dot

{

protected int height;

protected int width;

public int Height { get; set; }

public int Width { get; set; }

public Circle() : base()

{

var rand = new Random();

Height = rand.Next(300);

Width = rand.Next(300);

}

public Circle(int maxY, int maxX) : base(maxY, maxX)

{

var rand = new Random();

Height = rand.Next(maxY);

Width = rand.Next(maxX);

}

public Circle(int red, int green, int blue, int penWidth, int startX, int startY, int height, int width)

: base(red, green, blue, penWidth, startX, startY)

{

Height = height;

Width = width;

}

public Circle(Circle obj)

{

Height = obj.Height;

Width = obj.Width;

}

public override void Draw(Graphics canvas)

{

Pen pen = new Pen(System.Drawing.Color.FromArgb(Color.Red, Color.Green, Color.Blue), PenWidth);

canvas.DrawEllipse(pen, start.X, start.Y, Width, Height);

}

}

}

**Клас Ellipse**

using System;

using System.Drawing;

namespace ClassLibraryFigure

{

public class Ellipse : Circle

{

public Ellipse() : base()

{

}

public Ellipse(int maxY, int maxX) : base(maxY, maxX)

{

}

public Ellipse(int red, int green, int blue, int penWidth, int startX, int startY, int height, int width)

: base(red, green, blue, penWidth, startX, startY, height, width)

{

}

public override void Draw(Graphics canvas)

{

Pen pen = new Pen(System.Drawing.Color.FromArgb(Color.Red, Color.Green, Color.Blue), PenWidth);

canvas.DrawEllipse(pen, start.X, start.Y, Width, Height);

}

}

}

**Клас Rectangle**

using System;

using System.Drawing;

namespace ClassLibraryFigure

{

public class Rectangle : Line

{

public Rectangle() : base()

{

}

public Rectangle(int maxY, int maxX) : base(maxY, maxX)

{

}

public Rectangle(int red, int green, int blue, int penWidth, int startX, int startY, int endX, int endY)

: base(red, green, blue, penWidth, startX, startY, endX, endY)

{

}

public override void Draw(Graphics canvas)

{

Pen pen = new Pen(System.Drawing.Color.FromArgb(Color.Red, Color.Green, Color.Blue), PenWidth);

canvas.DrawRectangle(pen, start.X, start.Y, Math.Abs(start.X - end.X), Math.Abs(start.Y - end.Y));

}

}

}

Результат виконання програми:

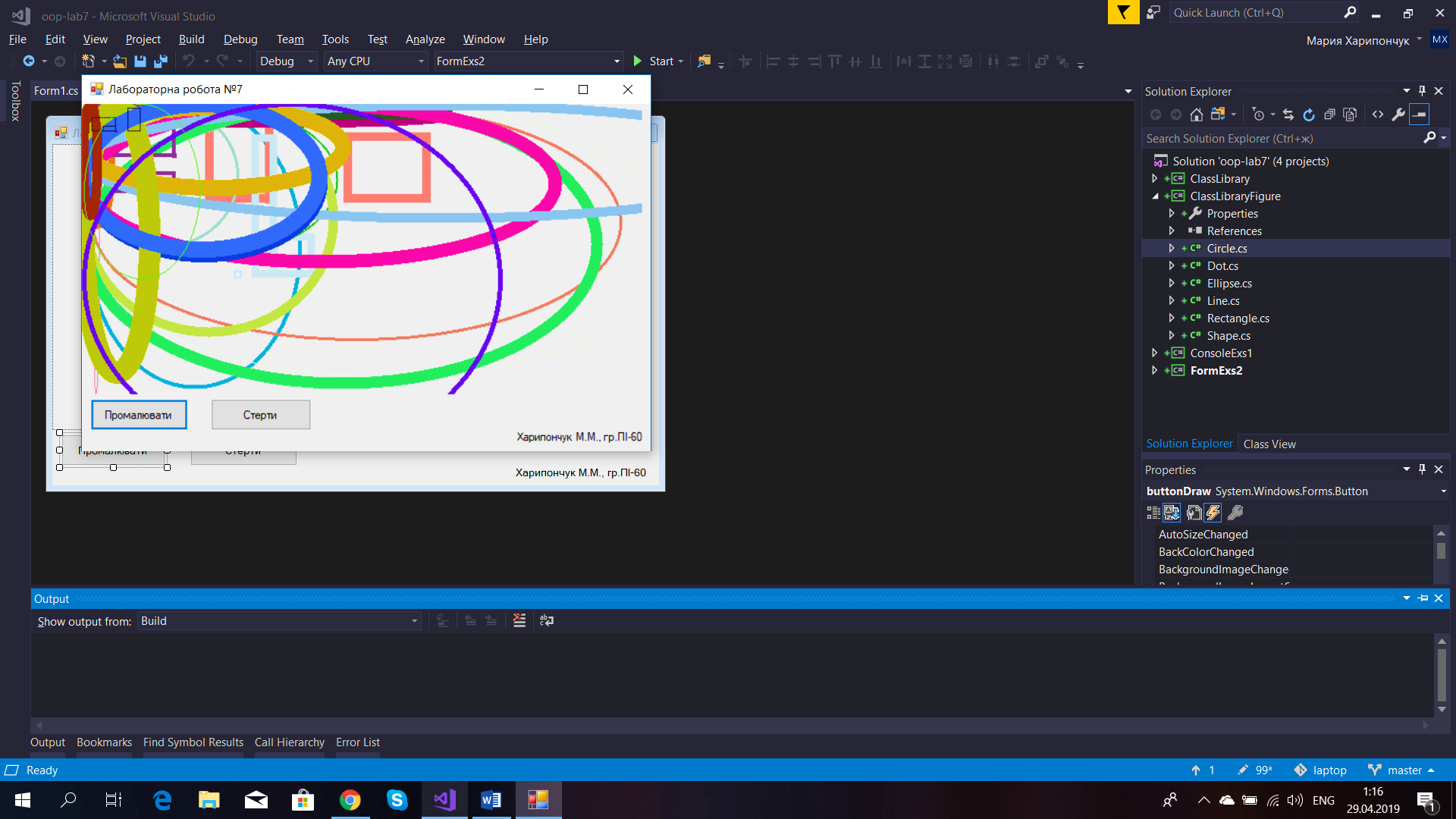


Рис. 2. Результат виконання роботи програми

***Висновок***: на даній лабораторній роботі було засвоєно принципи реалізації класів за допомогою наслідування.