**ЛАБОРАТОРНА РОБОТА № 8**

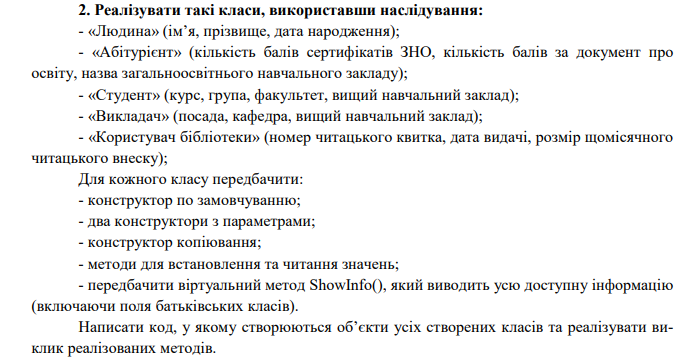
**Варіант 2**

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

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

**Хід роботи:**

**Завдання 1**:

****

**Лістинг класу People**:

namespace lab\_8

{

public class People

{

public string name;

public string surname;

public string date;

public People()

{

name = "Владислав";

surname = "Маньківський";

date = "24.03.2004";

}

public People(string name, string surname, string date)

{

this.name = name;

this.surname = surname;

this.date = date;

}

public People(People prev)

{

name = prev.name;

surname = prev.surname;

date = prev.date;

}

public string Name

{

get { return name; }

set { name = value; }

}

public string Surname

{

get { return surname; }

set { surname = value; }

}

public string Date

{

get { return date; }

set { date = value; }

}

public virtual void ShowInfo()

{

Console.WriteLine($"Ім'я:{name}\tПрізвище:{surname}\tДата.нар:{date}");

}

}

}

**Лістинг класу Applicant**:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab\_8

{

public class Applicant: People

{

public int math;

public int ukr;

public int angl;

public int avg;

public string zaklad;

public Applicant()

{

math = 182;

ukr = 189;

angl = 145;

avg = 9;

zaklad = "Ліцей №24";

}

public Applicant(int math, int ukr, int angl, int avg, string zaklad)

{

this.math = math;

this.ukr = ukr;

this.angl = angl;

this.avg = avg;

this.zaklad = zaklad;

}

public Applicant(Applicant prev)

{

math = prev.math;

ukr = prev.ukr;

angl = prev.angl;

avg = prev.avg;

zaklad = prev.zaklad;

}

public int Math

{

get { return math; }

set { math = value; }

}

public int Ukr

{

get { return ukr; }

set { ukr = value; }

}

public int Angl

{

get { return angl; }

set { angl = value; }

}

public int Avg

{

get { return avg; }

set { avg = value; }

}

public string Zaklad

{

get { return zaklad; }

set { zaklad = value; }

}

public virtual void ShowInfo()

{

Console.WriteLine($"---Студент---\nІм'я:{name}\tПрізвище:{surname}\tДата.нар:{date}");

Console.WriteLine($"Математика:{math}\tУкр.мова:{ukr}\tАнгл.мова:{angl}\tБал атестата:{avg}\tЗаклад:{zaklad}");

}

}

}

**Лістинг класу Student**:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab\_8

{

public class Student:Applicant

{

public int course;

public string group;

public string faculty;

public string vnz;

public Student()

{

course = 1;

group = "ВТ";

faculty = "ФІКТ";

vnz = "Житомирська політехніка";

}

public Student(int course, string group, string faculty, string vnz)

{

this.course = course;

this.group = group;

this.faculty = faculty;

this.vnz = vnz;

}

public Student(Student prev)

{

course = prev.course;

group = prev.group;

faculty = prev.faculty;

vnz = prev.vnz;

}

public int Course

{

get { return course; }

set { course = value; }

}

public string Group

{

get { return group; }

set { group = value; }

}

public string Faculty

{

get { return faculty; }

set { faculty = value; }

}

public string Vnz

{

get { return vnz; }

set { vnz = value; }

}

public virtual void ShowInfo()

{

Console.WriteLine($"---Студент---\nІм'я:{name}\tПрізвище:{surname}\tДата.нар:{date}");

Console.WriteLine($"Курс:{course}\tГрупа:{group}\tФакультет:{faculty}\tВНЗ:{vnz}");

Console.WriteLine($"Математика:{math}\tУкр.мова:{ukr}\tАнгл.мова:{angl}\tБал атестата:{avg}\tЗаклад:{zaklad}");

}

}

}

**Лістинг класу Biblioteka**:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab\_8

{

public class Biblioteka: Student

{

public int id;

public string data;

public int rozmir;

public Biblioteka()

{

id = 5;

data = "09.06.2022";

rozmir = 50;

}

public Biblioteka(int id, string data, int rozmir)

{

this.id = id;

this.data = data;

this.rozmir = rozmir;

}

public Biblioteka(Biblioteka prev)

{

id = prev.id;

data = prev.data;

rozmir = prev.rozmir;

name = prev.name;

surname = prev.surname;

date = prev.data;

course = prev.course;

group = prev.group;

faculty = prev.faculty;

vnz = prev.vnz;

math = prev.math;

ukr = prev.ukr;

angl = prev.angl;

avg = prev.avg;

zaklad = prev.zaklad;

}

public int Id

{

get { return id; }

set { id = value; }

}

public string Data

{

get { return data; }

set { data = value; }

}

public int Rozmir

{

get { return rozmir; }

set { rozmir = value; }

}

public virtual void ShowInfo()

{

Console.WriteLine($"---Користувач бібліотеки---\nІм'я:{name}\tПрізвище:{surname}\tДата.нар:{date}");

Console.WriteLine($"Курс:{course}\tГрупа:{group}\tФакультет:{faculty}\tВНЗ:{vnz}");

Console.WriteLine($"Математика:{math}\tУкр.Мова:{ukr}\tАнгл.Мова:{angl}\tБал атестата:{avg}\tШкола:{zaklad}");

Console.WriteLine($"Номер:{id}\tДата видачі:{data}\tВнесок:{rozmir}");

}

}

}

**Лістинг класу Prepod**:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab\_8

{

public class Prepod:People

{

public string posada;

public string kafedra;

public string vvnz;

public Prepod()

{

posada = "викладач програмування";

kafedra = "компютерні науки";

vvnz = "Житомирська політехніка";

}

public Prepod(string posada, string kafedra, string vvnz)

{

this.posada = posada;

this.kafedra = kafedra;

this.vvnz = vvnz;

}

public Prepod(Prepod prev)

{

posada = prev.posada;

kafedra = prev.kafedra;

vvnz = prev.vvnz;

}

public string Posada

{

get { return posada; }

set { posada = value; }

}

public string Kafedra

{

get { return kafedra; }

set { kafedra = value; }

}

public string Vvnz

{

get { return vvnz; }

set { vvnz = value; }

}

public virtual void ShowInfo()

{

Console.WriteLine($"---Викладач---\nІм'я:{name}\tПрізвище:{surname}\tДата.нар:{date}");

Console.WriteLine($"Посада:{posada}\tКафедра:{kafedra}\tВНЗ:{vvnz}");

}

}

}

**Лістинг класу Program**:

namespace lab\_8

{

class Program

{

static void Main()

{

Console.OutputEncoding = System.Text.Encoding.Unicode;

Console.InputEncoding = System.Text.Encoding.Unicode;

int fl;

Applicant appl = new Applicant();

Student stud = new Student();

Biblioteka bibl = new Biblioteka();

Prepod prep = new Prepod();

Console.WriteLine("---Абітурієнт---\n[1]По замовчуванню\n[2]Ввести дані");

while (!int.TryParse(Console.ReadLine(), out fl))

{

Console.WriteLine("Помилка введення!");

}

if(fl == 2)

{

Console.Write("Ім'я:");

appl.Name = Console.ReadLine();

Console.Write("Прізвище:");

appl.Surname = Console.ReadLine();

Console.Write("Дата народження:");

appl.Date = Console.ReadLine();

Console.Write("Математика:");

while (!int.TryParse(Console.ReadLine(), out fl))

{

Console.WriteLine("Помилка введення!");

}

appl.Math = fl;

Console.Write("Укр.мова:");

while (!int.TryParse(Console.ReadLine(), out fl))

{

Console.WriteLine("Помилка введення!");

}

appl.Ukr = fl;

Console.Write("Англ.мова:");

while (!int.TryParse(Console.ReadLine(), out fl))

{

Console.WriteLine("Помилка введення!");

}

appl.Angl = fl;

Console.Write("Бал атестата:");

while (!int.TryParse(Console.ReadLine(), out fl))

{

Console.WriteLine("Помилка введення!");

}

appl.Avg = fl;

Console.Write("Навчальний заклад:");

appl.Zaklad = Console.ReadLine();

}

Console.WriteLine("[1]Вивести запис");

while (!int.TryParse(Console.ReadLine(), out fl))

{

Console.WriteLine("Помилка введення!");

}

if(fl == 1)

{

Console.Clear();

appl.ShowInfo();

}

Console.ReadKey();

Console.Clear();

Console.WriteLine("---Студент---\n[1]По замовчуванню\n[2]Ввести дані");

while (!int.TryParse(Console.ReadLine(), out fl))

{

Console.WriteLine("Помилка введення!");

}

if (fl == 2)

{

Console.Write("Ім'я:");

stud.Name = Console.ReadLine();

Console.Write("Прізвище:");

stud.Surname = Console.ReadLine();

Console.Write("Дата народження:");

stud.Date = Console.ReadLine();

Console.Write("Курс:");

while (!int.TryParse(Console.ReadLine(), out fl))

{

Console.WriteLine("Помилка введення!");

}

stud.course = fl;

Console.Write("Група:");

stud.Group = Console.ReadLine();

Console.Write("Факультет:");

stud.faculty = Console.ReadLine();

Console.Write("Вищий навчальний заклад:");

stud.Vnz = Console.ReadLine();

}

Console.WriteLine("[1]Вивести запис");

while (!int.TryParse(Console.ReadLine(), out fl))

{

Console.WriteLine("Помилка введення!");

}

if (fl == 1)

{

Console.Clear();

stud.ShowInfo();

}

Console.ReadKey();

Console.Clear();

Console.WriteLine("---Викладач---\n[1]По замовчуванню\n[2]Ввести дані");

while (!int.TryParse(Console.ReadLine(), out fl))

{

Console.WriteLine("Помилка введення!");

}

if (fl == 2)

{

Console.Write("Ім'я:");

prep.Name = Console.ReadLine();

Console.Write("Прізвище:");

prep.Surname = Console.ReadLine();

Console.Write("Дата народження:");

prep.Date = Console.ReadLine();

Console.Write("Посада:");

prep.posada = Console.ReadLine();

Console.Write("Кафедра:");

prep.kafedra = Console.ReadLine();

Console.Write("Вищий навчальний заклад:");

prep.vvnz = Console.ReadLine();

}

Console.WriteLine("[1]Вивести запис");

while (!int.TryParse(Console.ReadLine(), out fl))

{

Console.WriteLine("Помилка введення!");

}

if (fl == 1)

{

Console.Clear();

prep.ShowInfo();

}

Console.ReadKey();

Console.Clear();

Console.WriteLine("---Користувач бібліотеки---\n[1]По замовчуванню\n[2]Ввести дані");

while (!int.TryParse(Console.ReadLine(), out fl))

{

Console.WriteLine("Помилка введення!");

}

if (fl == 2)

{

Console.Write("Ім'я:");

bibl.Name = Console.ReadLine();

Console.Write("Прізвище:");

bibl.Surname = Console.ReadLine();

Console.Write("Дата народження:");

bibl.Date = Console.ReadLine();

Console.WriteLine("Номер:");

while (!int.TryParse(Console.ReadLine(), out fl))

{

Console.WriteLine("Помилка введення!");

}

bibl.Id = fl;

Console.Write("Дата видачі:");

bibl.Data = Console.ReadLine();

Console.Write("Розмір внеску:");

while (!int.TryParse(Console.ReadLine(), out fl))

{

Console.Write("Помилка введення!");

}

bibl.Rozmir = fl;

}

Console.WriteLine("[1]Вивести запис");

while (!int.TryParse(Console.ReadLine(), out fl))

{

Console.WriteLine("Помилка введення!");

}

if (fl == 1)

{

Console.Clear();

bibl.ShowInfo();

}

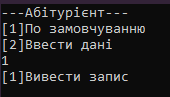
Console.ReadKey();

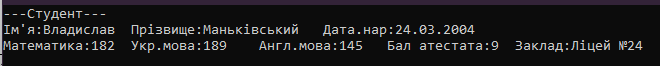
Console.Clear();

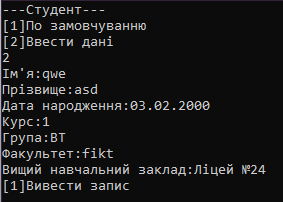
}

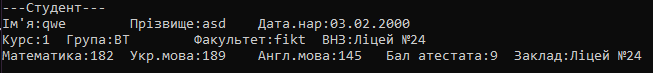
}

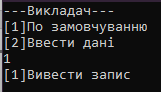
}

******

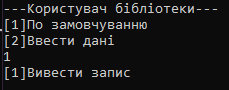
******

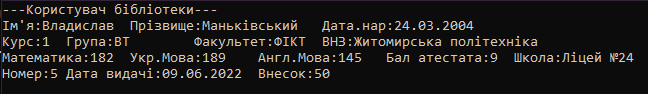
******

******

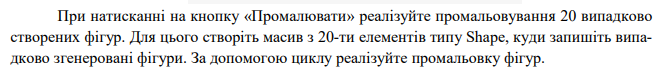
******

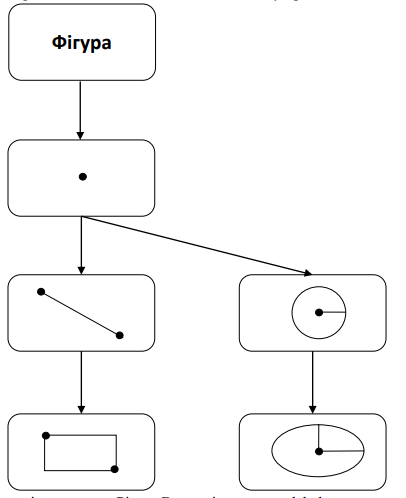
******

******

******

**Завдання 2**:

******

******

**Лістинг:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace task\_2

{

public partial class Form1 : Form

{

Random random = new Random();

public Form1()

{

InitializeComponent();

}

Graphics g;

Point[] point = new Point[100];

Rect[] rect = new Rect[100];

Line[] line = new Line[100];

Circ[] circ = new Circ[100];

Ell[] ell = new Ell[100];

private void Form1\_Load(object sender, EventArgs e)

{

FormBorderStyle = FormBorderStyle.FixedSingle;

MaximizeBox = false;

MinimizeBox = false;

}

private void button2\_Click(object sender, EventArgs e)

{

g.Clear(Color.White);

}

private void button1\_Click(object sender, EventArgs e)

{

int fl;

for (int i = 0; i < 20; i++)

{

g = pictureBox1.CreateGraphics();

fl = random.Next(1, 6);

if (fl == 1)

{

point[i] = new Point(RandX(), RandY());

point[i].Draw(g);

}

if (fl == 2)

{

line[i] = new Line(RandX(), RandY(), RandX(), RandY());

line[i].Draw(g);

}

if (fl == 3)

{

rect[i] = new Rect(RandX(), RandY());

rect[i].Draw(g);

}

if (fl == 4)

{

circ[i] = new Circ(RandX(), RandY());

circ[i].Draw(g);

}

if (fl == 5)

{

ell[i] = new Ell(RandX(), RandY());

ell[i].Draw(g);

}

}

}

public float RandX()

{

float x = random.Next(0, 800);

return x;

}

public float RandY()

{

float y = random.Next(0, 300);

return y;

}

public class Shape

{

protected float x;

protected float y;

protected float x1;

protected float y1;

public Shape(float x, float y)

{

this.x = x;

this.y = y;

}

public Shape(Shape shape)

{

this.x = shape.x;

this.y = shape.y;

}

public void Draw(Graphics g)

{

}

}

public class Point : Shape

{

public Point(float x, float y) : base(x, y)

{

this.x = x;

this.y = y;

}

public virtual void Draw(Graphics g)

{

g.DrawEllipse(Pens.Black, x, y, 10, 10);

}

}

public class Line : Point

{

public Line(float x, float y, float x1, float y1) : base(x, y)

{

this.x = x;

this.y = y;

this.x1 = x1;

this.y1 = y1;

}

public virtual void Draw(Graphics g)

{

g.DrawLine(Pens.Black, x, y, x1, y1);

}

}

public class Rect : Shape

{

public Rect(float x, float y) : base(x, y)

{

this.x = x;

this.y = y;

}

public virtual void Draw(Graphics g)

{

g.DrawRectangle(Pens.Black, x, y, 75, 75);

}

}

public class Circ : Shape

{

public Circ(float x, float y) : base(x, y)

{

this.x = x;

this.y = y;

}

public virtual void Draw(Graphics g)

{

g.DrawEllipse(Pens.Black, x, y, 50, 50);

}

}

public class Ell : Circ

{

public Ell(float x, float y) : base(x, y)

{

this.x = x;

this.y = y;

}

public virtual void Draw(Graphics g)

{

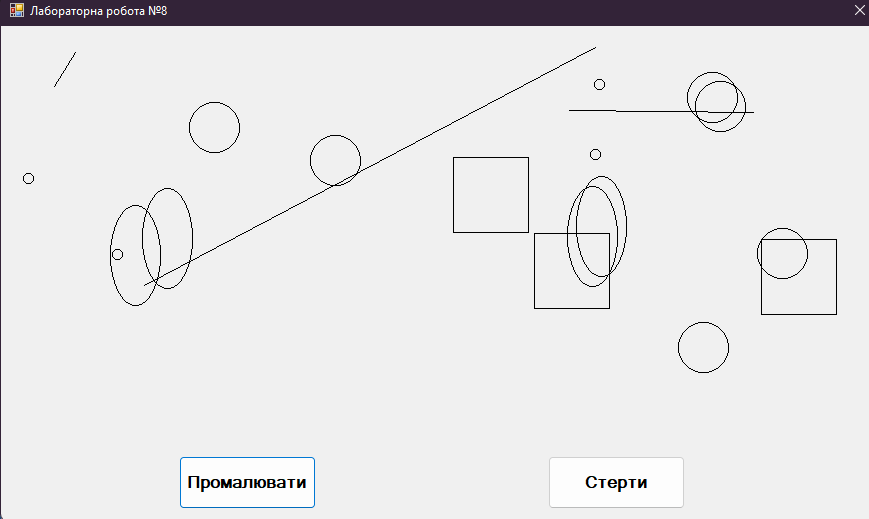
g.DrawEllipse(Pens.Black, x, y, 50, 100);

}

}

}

}

******

***Висновки:*** я засвоїв принципи реалізації класів за допомогою наслідування.