Міністерство освіти і науки України

Черкаський державний технологічний університет

Кафедра програмного забезпечення автоматизованих систем

ЗВІТ

про виконання лабораторної роботи № 2

з дисципліни:

«Об’єктно орієнтоване програмування»

|  |  |
| --- | --- |
| Перевірив:  викладач кафедри ПЗАС  Крайовий В. М .  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2016 р. | Виконав:  студент 2 курсу  групи ПЗ-154  Журавленко М. В. |

Черкаси 2016

# Лабораторна робота № 1

**Тема роботи:** Найпростіші програми на Java. Абстракція. Інкапсуляція. Створення і використання власного класу.

**Мета роботи:** Навчитись створювати власні класи, описувати поля та методи. Навчитись використовувати власні класи.

**Індивідуальне завдання:** Створити на мові Java клас, який описує поняття реального світу згідно з варіантом завдання. Клас повинен мати не менше 5 полів, що описують властивості данго поняття та не менше 3 методів, які описують його поведінку. Методи повинні працювати з полями, читати або записувати їх; всі поля повинні бути задіяні в методах. Імена полів повинні починатись з іменника або прикметника, методів - з дієслова. Створити програму, яка створює вікно з чотирма кнопками. При натисненні на першу кнопку повинен створюватись об’єкт нашого класу, при натисненні на кожну з інших кнопок повинен запускатись відповідний метод нашого класу

Варіанти (поняття реального світу для створення класу).

5. Квітка.

**Текст програми на Java:**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.util.Random;

//---створення головного класу

class main {

public static void main(String[] args) {

//---Підключення класів

Random random = new Random();

Flower\_properties flower\_properties\_cls = new Flower\_properties();

//---Створення головної форми

JFrame Flower = new JFrame("Flower");

Flower.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Flower.setVisible(true);

JButton create\_flower = new JButton("Create Flower");

create\_flower.setFont(new Font("Verdana", Font.PLAIN, 15));

create\_flower.setPreferredSize(new Dimension(100, 50));

GridLayout flower\_grid = new GridLayout(1, 1);

Flower.setLayout(flower\_grid);

Flower.getContentPane().add(create\_flower);

Flower.setSize(300, 70);

Flower.setResizable(false);

//---створення форми для вибору квітки

JFrame Flower\_Choice = new JFrame("Flower Choice");

Flower\_Choice.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Flower\_Choice.setVisible(false);

JButton flower\_random = new JButton("Create Random Flower");

flower\_random.setFont(new Font("Verdana", Font.PLAIN, 15));

flower\_random.setPreferredSize(new Dimension(100, 50));

JButton flower\_from\_list = new JButton("Create Flower From List");

flower\_from\_list.setFont(new Font("Verdana", Font.PLAIN, 15));

flower\_from\_list.setPreferredSize(new Dimension(100, 50));

GridLayout create\_flower\_grid = new GridLayout(1, 2);

Flower\_Choice.setLayout(create\_flower\_grid);

Flower\_Choice.getContentPane().add(flower\_random);

Flower\_Choice.getContentPane().add(flower\_from\_list);

Flower\_Choice.setSize(400, 70);

Flower\_Choice.setResizable(false);

//---створення форми для вибору квітки зі списку

JFrame Flower\_Choice\_From\_List = new JFrame("Flower Choice From List");

Flower\_Choice.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Flower\_Choice.setVisible(false);

JComboBox<String> flower\_list = new JComboBox<>(flower\_properties\_cls.get\_array\_name\_of\_flowers());

flower\_list.setFont(new Font("Verdana", Font.PLAIN, 15));

flower\_list.setPreferredSize(new Dimension(100, 50));

JButton ok = new JButton("Ok");

ok.setFont(new Font("Verdana", Font.PLAIN, 15));

ok.setPreferredSize(new Dimension(100, 50));

GridLayout flower\_list\_grid = new GridLayout(2, 1);

Flower\_Choice\_From\_List.setLayout(flower\_list\_grid);

Flower\_Choice\_From\_List.getContentPane().add(flower\_list);

Flower\_Choice\_From\_List.getContentPane().add(ok);

Flower\_Choice\_From\_List.setSize(400, 140);

Flower\_Choice\_From\_List.setResizable(false);

//---творення форми для квітки

JFrame Current\_Flower = new JFrame("Current Flower");

Current\_Flower.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Current\_Flower.setVisible(false);

JTextField current\_flower\_name = new JTextField();

current\_flower\_name.setFont(new Font("Verdana", Font.PLAIN, 15));

JTextField current\_flower\_number\_of\_blades = new JTextField();

current\_flower\_number\_of\_blades.setFont(new Font("Verdana", Font.PLAIN, 15));

JTextField current\_flower\_height\_of\_flowers = new JTextField();

current\_flower\_height\_of\_flowers.setFont(new Font("Verdana", Font.PLAIN, 15));

JTextField current\_flower\_make\_tea = new JTextField();

current\_flower\_make\_tea.setFont(new Font("Verdana", Font.PLAIN, 15));

JTextField current\_flower\_color\_of\_blades = new JTextField();

current\_flower\_color\_of\_blades.setFont(new Font("Verdana", Font.PLAIN, 15));

JTextField current\_flower\_healing\_porperties = new JTextField();

current\_flower\_healing\_porperties.setFont(new Font("Verdana", Font.PLAIN, 15));

JTextField current\_flower\_the\_place\_where\_they\_grows = new JTextField();

current\_flower\_the\_place\_where\_they\_grows.setFont(new Font("Verdana", Font.PLAIN, 15));

JTextField current\_flower\_familia\_of\_flowers = new JTextField();

current\_flower\_familia\_of\_flowers.setFont(new Font("Verdana", Font.PLAIN, 15));

JTextField current\_flower\_form\_of\_blades = new JTextField();

current\_flower\_form\_of\_blades.setFont(new Font("Verdana", Font.PLAIN, 15));

JTextField current\_flower\_domain\_of\_flowers = new JTextField();

current\_flower\_domain\_of\_flowers.setFont(new Font("Verdana", Font.PLAIN, 15));

JTextField current\_flower\_regnum\_of\_flowers = new JTextField();

current\_flower\_regnum\_of\_flowers.setFont(new Font("Verdana", Font.PLAIN, 15));

JTextField current\_flower\_division\_of\_flowers = new JTextField();

current\_flower\_division\_of\_flowers.setFont(new Font("Verdana", Font.PLAIN, 15));

JTextField current\_flower\_classis\_of\_flowers = new JTextField();

current\_flower\_classis\_of\_flowers.setFont(new Font("Verdana", Font.PLAIN, 15));

JTextField current\_flower\_ordo\_of\_flowers = new JTextField();

current\_flower\_ordo\_of\_flowers.setFont(new Font("Verdana", Font.PLAIN, 15));

JTextField current\_flower\_genus\_of\_flowers = new JTextField();

current\_flower\_genus\_of\_flowers.setFont(new Font("Verdana", Font.PLAIN, 15));

JButton current\_flower\_plant\_flower = new JButton("Plant Flower");

current\_flower\_plant\_flower.setFont(new Font("Verdana", Font.PLAIN, 15));

JButton current\_flower\_watered\_flower = new JButton("Watered Flower");

current\_flower\_watered\_flower.setFont(new Font("Verdana", Font.PLAIN, 15));

JButton current\_flower\_tear\_off\_petal = new JButton("Tear Off Petal");

current\_flower\_tear\_off\_petal.setFont(new Font("Verdana", Font.PLAIN, 15));

JButton current\_flower\_pick\_the\_flower = new JButton("Pick the flower");

current\_flower\_pick\_the\_flower.setFont(new Font("Verdana", Font.PLAIN, 15));

JTextField current\_flower\_water\_level = new JTextField();

current\_flower\_water\_level.setFont(new Font("Verdana", Font.PLAIN, 15));

JTextField current\_flower\_level\_of\_nutrients = new JTextField();

current\_flower\_level\_of\_nutrients.setFont(new Font("Verdana", Font.PLAIN, 15));

JTextField current\_flower\_new\_number\_of\_blades = new JTextField();

current\_flower\_new\_number\_of\_blades.setFont(new Font("Verdana", Font.PLAIN, 15));

JTextField current\_flower\_is\_plant\_flower = new JTextField();

current\_flower\_is\_plant\_flower.setFont(new Font("Verdana", Font.PLAIN, 15));

JPanel current\_flower\_info\_jpanel = new JPanel(new GridLayout(5, 3));

current\_flower\_info\_jpanel.setPreferredSize(new Dimension(900, 800));

current\_flower\_info\_jpanel.add(current\_flower\_name);

current\_flower\_info\_jpanel.add(current\_flower\_number\_of\_blades);

current\_flower\_info\_jpanel.add(current\_flower\_height\_of\_flowers);

current\_flower\_info\_jpanel.add(current\_flower\_make\_tea);

current\_flower\_info\_jpanel.add(current\_flower\_color\_of\_blades);

current\_flower\_info\_jpanel.add(current\_flower\_healing\_porperties);

current\_flower\_info\_jpanel.add(current\_flower\_the\_place\_where\_they\_grows);

current\_flower\_info\_jpanel.add(current\_flower\_familia\_of\_flowers);

current\_flower\_info\_jpanel.add(current\_flower\_form\_of\_blades);

current\_flower\_info\_jpanel.add(current\_flower\_domain\_of\_flowers);

current\_flower\_info\_jpanel.add(current\_flower\_regnum\_of\_flowers);

current\_flower\_info\_jpanel.add(current\_flower\_division\_of\_flowers);

current\_flower\_info\_jpanel.add(current\_flower\_classis\_of\_flowers);

current\_flower\_info\_jpanel.add(current\_flower\_ordo\_of\_flowers);

current\_flower\_info\_jpanel.add(current\_flower\_genus\_of\_flowers);

JPanel current\_flower\_action\_jpanel = new JPanel(new GridLayout(4, 1));

current\_flower\_action\_jpanel.add(current\_flower\_plant\_flower);

current\_flower\_action\_jpanel.add(current\_flower\_watered\_flower);

current\_flower\_action\_jpanel.add(current\_flower\_tear\_off\_petal);

current\_flower\_action\_jpanel.add(current\_flower\_pick\_the\_flower);

JPanel current\_flower\_action\_info\_jpanel = new JPanel(new GridLayout(4, 1));

current\_flower\_action\_info\_jpanel.add(current\_flower\_water\_level);

current\_flower\_action\_info\_jpanel.add(current\_flower\_level\_of\_nutrients);

current\_flower\_action\_info\_jpanel.add(current\_flower\_new\_number\_of\_blades);

current\_flower\_action\_info\_jpanel.add(current\_flower\_is\_plant\_flower);

Container current\_flower\_container = Current\_Flower.getContentPane();

current\_flower\_container.setLayout(new GridLayout(3, 1));

current\_flower\_container.add(current\_flower\_info\_jpanel);

current\_flower\_container.add(current\_flower\_action\_jpanel);

current\_flower\_container.add(current\_flower\_action\_info\_jpanel);

Current\_Flower.setSize(900, 700);

Current\_Flower.setResizable(false);

//---Створення обробників натискання кнопо

create\_flower.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

Flower\_Choice.setVisible(true);

}

}

);

flower\_random.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

int rnd = random.nextInt(5);

flower\_properties\_cls.rrr = flower\_properties\_cls.get\_number\_of\_blades(rnd);

current\_flower\_name.setText("the name of flower - \n" + flower\_properties\_cls.get\_name\_of\_flowers(rnd));

current\_flower\_number\_of\_blades.setText("the number of blades - \n" + Integer.toString(flower\_properties\_cls.get\_number\_of\_blades(rnd)));

current\_flower\_height\_of\_flowers.setText("the height of flower - \n" + Integer.toString(flower\_properties\_cls.get\_height\_of\_flowers(rnd)));

current\_flower\_make\_tea.setText("can i make a tea with flower ? - \n" + Boolean.toString(flower\_properties\_cls.get\_make\_tea(rnd)));

current\_flower\_color\_of\_blades.setText("the color of flower blades - \n" + flower\_properties\_cls.get\_color\_of\_blades(rnd));

current\_flower\_healing\_porperties.setText("the healing porperties of this flower - \n" + flower\_properties\_cls.get\_healing\_porperties(rnd));

current\_flower\_the\_place\_where\_they\_grows.setText("the place where they grows - \n" + flower\_properties\_cls.get\_the\_place\_where\_they\_grows(rnd));

current\_flower\_familia\_of\_flowers.setText("the familia of flowers - \n" + flower\_properties\_cls.get\_familia\_of\_flowers(rnd));

current\_flower\_form\_of\_blades.setText("the form of blades - \n" + flower\_properties\_cls.get\_form\_of\_blades(rnd));

current\_flower\_domain\_of\_flowers.setText("the domain of flowers - \n" + flower\_properties\_cls.get\_domain\_of\_flowers(rnd));

current\_flower\_regnum\_of\_flowers.setText("the regnum of flowers - \n" + flower\_properties\_cls.get\_regnum\_of\_flowers(rnd));

current\_flower\_division\_of\_flowers.setText("the division of flowers - \n" + flower\_properties\_cls.get\_division\_of\_flowers(rnd));

current\_flower\_classis\_of\_flowers.setText("the classis of flowers - \n" + flower\_properties\_cls.get\_classis\_of\_flowers(rnd));

current\_flower\_ordo\_of\_flowers.setText("the ordo of flowers - \n" + flower\_properties\_cls.get\_ordo\_of\_flowers(rnd));

current\_flower\_genus\_of\_flowers.setText("the genus of flowers - \n" + flower\_properties\_cls.get\_genus\_of\_flowers(rnd));

Current\_Flower.setVisible(true);

}

}

);

flower\_from\_list.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

Flower\_Choice.setVisible(false);

Flower\_Choice\_From\_List.setVisible(true);

}

}

);

ok.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

int rnd = flower\_list.getSelectedIndex();

flower\_properties\_cls.rrr = flower\_properties\_cls.get\_number\_of\_blades(rnd);

current\_flower\_name.setText("the name of flower - \n" + flower\_properties\_cls.get\_name\_of\_flowers(rnd));

current\_flower\_number\_of\_blades.setText("the number of blades - \n" + Integer.toString(flower\_properties\_cls.get\_number\_of\_blades(rnd)));

current\_flower\_height\_of\_flowers.setText("the height of flower - \n" + Integer.toString(flower\_properties\_cls.get\_height\_of\_flowers(rnd)));

current\_flower\_make\_tea.setText("can i make a tea with flower ? - \n" + Boolean.toString(flower\_properties\_cls.get\_make\_tea(rnd)));

current\_flower\_color\_of\_blades.setText("the color of flower blades - \n" + flower\_properties\_cls.get\_color\_of\_blades(rnd));

current\_flower\_healing\_porperties.setText("the healing porperties of this flower - \n" + flower\_properties\_cls.get\_healing\_porperties(rnd));

current\_flower\_the\_place\_where\_they\_grows.setText("the place where they grows - \n" + flower\_properties\_cls.get\_the\_place\_where\_they\_grows(rnd));

current\_flower\_familia\_of\_flowers.setText("the familia of flowers - \n" + flower\_properties\_cls.get\_familia\_of\_flowers(rnd));

current\_flower\_form\_of\_blades.setText("the form of blades - \n" + flower\_properties\_cls.get\_form\_of\_blades(rnd));

current\_flower\_domain\_of\_flowers.setText("the domain of flowers - \n" + flower\_properties\_cls.get\_domain\_of\_flowers(rnd));

current\_flower\_regnum\_of\_flowers.setText("the regnum of flowers - \n" + flower\_properties\_cls.get\_regnum\_of\_flowers(rnd));

current\_flower\_division\_of\_flowers.setText("the division of flowers - \n" + flower\_properties\_cls.get\_division\_of\_flowers(rnd));

current\_flower\_classis\_of\_flowers.setText("the classis of flowers - \n" + flower\_properties\_cls.get\_classis\_of\_flowers(rnd));

current\_flower\_ordo\_of\_flowers.setText("the ordo of flowers - \n" + flower\_properties\_cls.get\_ordo\_of\_flowers(rnd));

current\_flower\_genus\_of\_flowers.setText("the genus of flowers - \n" + flower\_properties\_cls.get\_genus\_of\_flowers(rnd));

Current\_Flower.setVisible(true);

}

}

);

current\_flower\_plant\_flower.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

flower\_properties\_cls.plant\_flower(current\_flower\_is\_plant\_flower, current\_flower\_new\_number\_of\_blades, current\_flower\_level\_of\_nutrients, current\_flower\_water\_level, Current\_Flower);

}

}

);

current\_flower\_watered\_flower.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

flower\_properties\_cls.watered\_flower(current\_flower\_water\_level, current\_flower\_level\_of\_nutrients, Current\_Flower);

}

}

);

current\_flower\_tear\_off\_petal.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

flower\_properties\_cls.tear\_off\_petal(current\_flower\_new\_number\_of\_blades, Current\_Flower);

}

}

);

current\_flower\_pick\_the\_flower.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

flower\_properties\_cls.pick\_the\_flower(current\_flower\_is\_plant\_flower, current\_flower\_new\_number\_of\_blades, current\_flower\_level\_of\_nutrients, current\_flower\_water\_level, Current\_Flower);

}

}

);

}

}

//---класс з властивостями і методами

class Flower\_properties {

//---Оголошення полів

String[] name\_of\_flowers = {

"Aster",

"Begonia",

"Buttercup",

"Chrysanthemum",

"Crocus"

};

int[] number\_of\_blades = {13, 19, 5, 40, 6};

int[] height\_of\_flowers = {27, 45, 35, 65, 10};

boolean[] make\_tea = {

true,

true,

false,

false,

true

};

String[] color\_of\_blades = {

"white",

"red",

"yellow",

"pink",

"white"

};

String[] healing\_porperties = {

"itch, inflammation, flu",

"soothes and relieves stress",

"no",

"cold",

"cold"

};

String[] the\_place\_where\_they\_grows = {

"steppe",

"wet forest",

"moderate and cold climate",

"Japan",

"Ukraine"

};

String[] form\_of\_blades = {

"oval",

"circle",

"convex",

"oval",

"oval"

};

String[] familia\_of\_flowers = {

"Asteraceae",

"Begoniaceae",

"Ranunculaceae",

"Asteraceae",

"Iridaceae"

};

String[] domain\_of\_flowers = {

"Eukaryota",

"---",

"Eukaryota",

"Eukaryota",

"Eukaryota"

};

String[] regnum\_of\_flowers = {

"Viridiplantae",

"Viridiplantae",

"Viridiplantae",

"Viridiplantae",

"Viridiplantae"

};

String[] division\_of\_flowers = {

"Streptophyta",

"Streptophyta",

"Streptophyta",

"Streptophyta",

"Streptophyta"

};

String[] classis\_of\_flowers = {

"---",

"---",

"Eudicots",

"Eudicots",

"Liliopsida"

};

String[] ordo\_of\_flowers = {

"Asterales",

"Cucurbitales",

"Ranunculales",

"Asterales",

"Asparagales",

};

String[] genus\_of\_flowers = {

"Aster",

"Begonia",

"Ranunculus",

"Chrysanthemum",

"Crocus"

};

int water\_level = 0;

int level\_of\_nutrients = 0;

int new\_number\_of\_blades = 0;

boolean is\_plant\_flower = false;

int rrr = 0;

//---Оголошення методів

void change\_water\_level(int i) {

water\_level = water\_level + i;

}

void change\_level\_of\_nutrients(int i) {

level\_of\_nutrients = level\_of\_nutrients + i;

}

void change\_new\_number\_of\_blades(int i) {

if (new\_number\_of\_blades != 0) {

new\_number\_of\_blades = new\_number\_of\_blades - 1;

}

}

void change\_is\_plant\_flower(boolean tr) {

is\_plant\_flower = tr;

}

void plant\_flower(JTextField cur\_flower\_is\_plant\_flower, JTextField cur\_flower\_new\_number\_of\_blades, JTextField cur\_flower\_level\_of\_nutrients, JTextField cur\_flower\_water\_level, JFrame Cur\_Flower) {

change\_is\_plant\_flower(true);

change\_water\_level(15);

change\_level\_of\_nutrients(10);

new\_number\_of\_blades = rrr;

cur\_flower\_is\_plant\_flower.setText("Is plant flower - " + Boolean.toString(is\_plant\_flower));

cur\_flower\_new\_number\_of\_blades.setText("Number of blades - " + Integer.toString(new\_number\_of\_blades));

cur\_flower\_water\_level.setText("Water level - " + Integer.toString(water\_level));

cur\_flower\_level\_of\_nutrients.setText("Level of nutriens - " + Integer.toString(level\_of\_nutrients));

}

void watered\_flower(JTextField cur\_flower\_water\_level, JTextField cur\_flower\_level\_of\_nutrients, JFrame Current\_Flower) {

if (is\_plant\_flower) {

change\_water\_level(15);

change\_level\_of\_nutrients(12);

cur\_flower\_water\_level.setText("Water level - " + Integer.toString(water\_level));

cur\_flower\_level\_of\_nutrients.setText("Level of nutriens - " + Integer.toString(level\_of\_nutrients));

}

}

void tear\_off\_petal(JTextField cur\_flower\_new\_number\_of\_blades, JFrame Current\_Flower) {

if (is\_plant\_flower) {

change\_new\_number\_of\_blades(1);

cur\_flower\_new\_number\_of\_blades.setText("Number of blades - " + Integer.toString(new\_number\_of\_blades));

}

}

void pick\_the\_flower(JTextField cur\_flower\_is\_plant\_flower, JTextField cur\_flower\_new\_number\_of\_blades, JTextField cur\_flower\_level\_of\_nutrients, JTextField cur\_flower\_water\_level, JFrame Cur\_Flower) {

if (is\_plant\_flower) {

change\_is\_plant\_flower(false);

water\_level = 0;

level\_of\_nutrients = 0;

new\_number\_of\_blades = 0;

cur\_flower\_is\_plant\_flower.setText("Is plant flower - " + Boolean.toString(is\_plant\_flower));

cur\_flower\_new\_number\_of\_blades.setText("Number of blades - " + Integer.toString(new\_number\_of\_blades));

cur\_flower\_water\_level.setText("Water level - " + Integer.toString(water\_level));

cur\_flower\_level\_of\_nutrients.setText("Level of nutriens - " + Integer.toString(level\_of\_nutrients));

}

}

String[] get\_array\_name\_of\_flowers() {

return name\_of\_flowers;

}

String get\_name\_of\_flowers(int i) {

return name\_of\_flowers[i];

}

int get\_number\_of\_blades(int i) {

return number\_of\_blades[i];

}

int get\_height\_of\_flowers(int i) {

return height\_of\_flowers[i];

}

boolean get\_make\_tea(int i) {

return make\_tea[i];

}

String get\_color\_of\_blades(int i) {

return color\_of\_blades[i];

}

String get\_healing\_porperties(int i) {

return healing\_porperties[i];

}

String get\_the\_place\_where\_they\_grows(int i) {

return the\_place\_where\_they\_grows[i];

}

String get\_form\_of\_blades(int i) {

return form\_of\_blades[i];

}

String get\_familia\_of\_flowers(int i) {

return familia\_of\_flowers[i];

}

String get\_domain\_of\_flowers(int i) {

return domain\_of\_flowers[i];

}

String get\_regnum\_of\_flowers(int i) {

return regnum\_of\_flowers[i];

}

String get\_division\_of\_flowers(int i) {

return division\_of\_flowers[i];

}

String get\_classis\_of\_flowers(int i) {

return classis\_of\_flowers[i];

}

String get\_ordo\_of\_flowers(int i) {

return ordo\_of\_flowers[i];

}

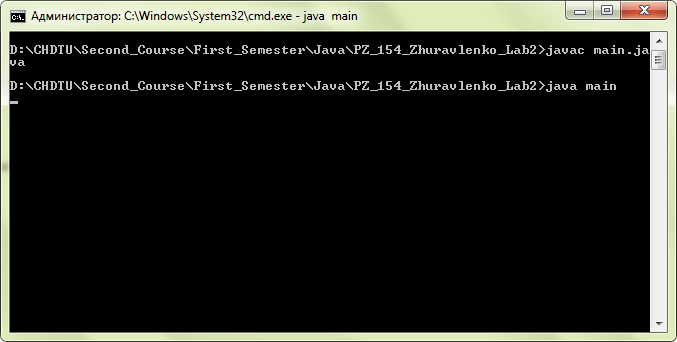
String get\_genus\_of\_flowers(int i) {

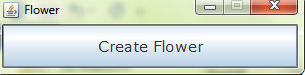
return genus\_of\_flowers[i];

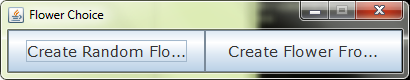
}

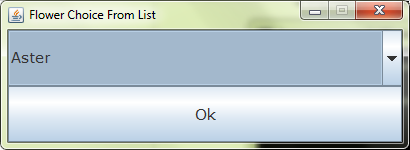
}

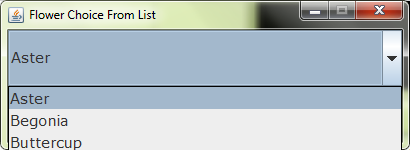
**Результати виконання програми на Java:**

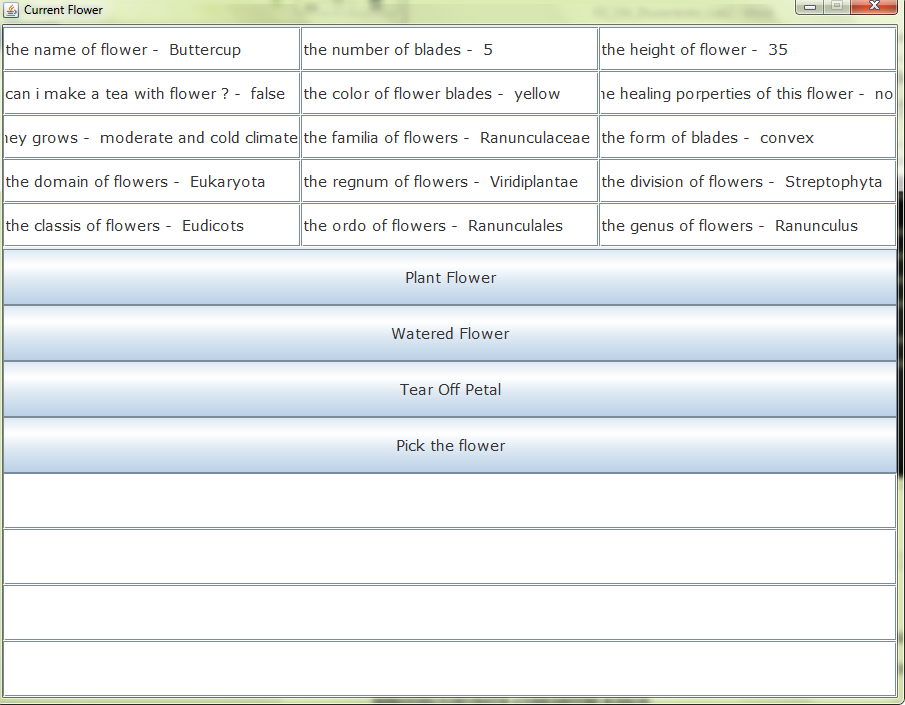


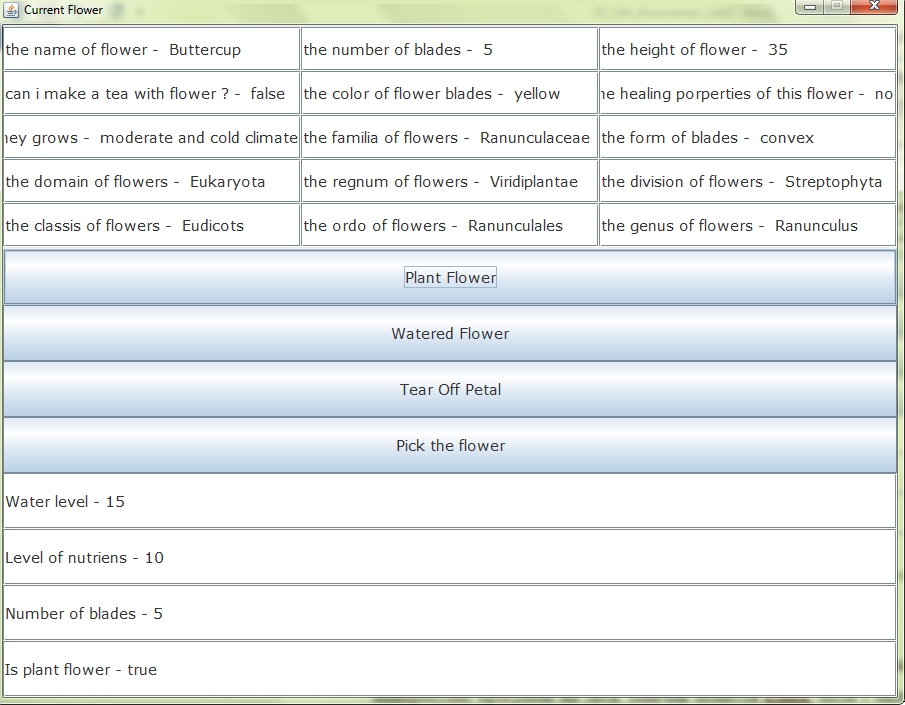


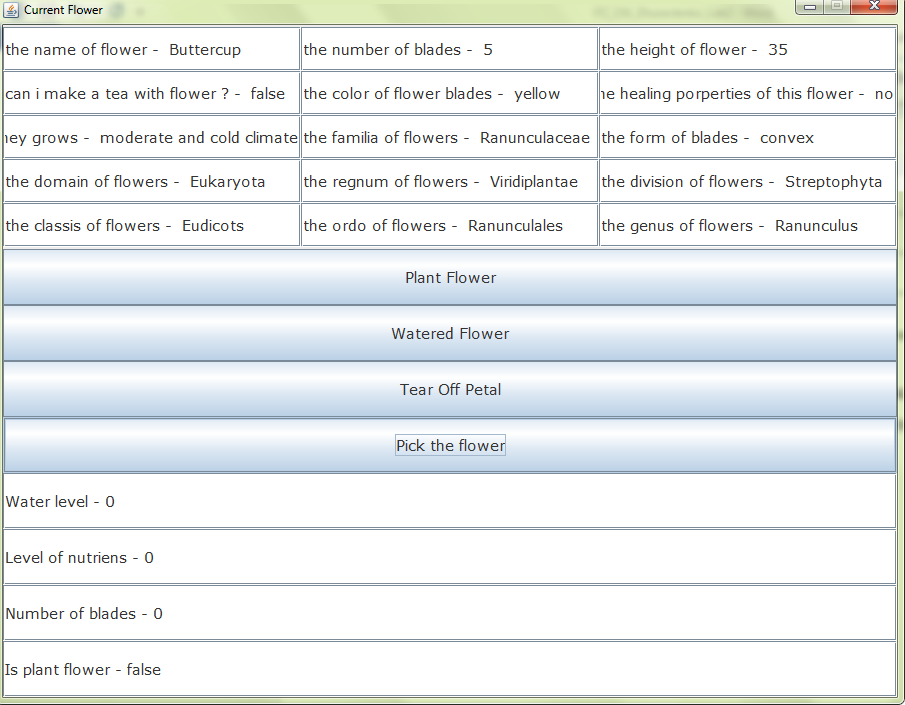












**Висновок:** Виконуючи цю лабораторну роботу, я навчився створювати власні класи, описувати поля та методи, а також використовувати їх.