Національний технічний університет України

‘Київський політехнічний інститут’

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

***Лабораторна робота №4***

*“ Відображення графічної інформації засобами Java (AWT)”*

*Виконав: Мроць Ю.Б.*

*Група: ІО-12*

*Номер зк: 1219*

*Перевірив: Абу Усбах О.Н.*

*Київ — 2013*

***Мета***

Ознайомлення з моделлю малювання в Java. Отримання базових навичок з використання засобів відображення графічної інформації в бібліотеці AWT. Розробка класу візуалізації табличних даних.

***Завдання***

1. Вивчити принципи побудови графічного інтерфейсу. Вільно володіти поняттями компонента, контейнера та графічного контексту.

2. Ознайомитись з бібліотеками графічних інтерфейсів AWT та SWING. Вільно володіти поняттями “важкого” та “легкого” компонентів.

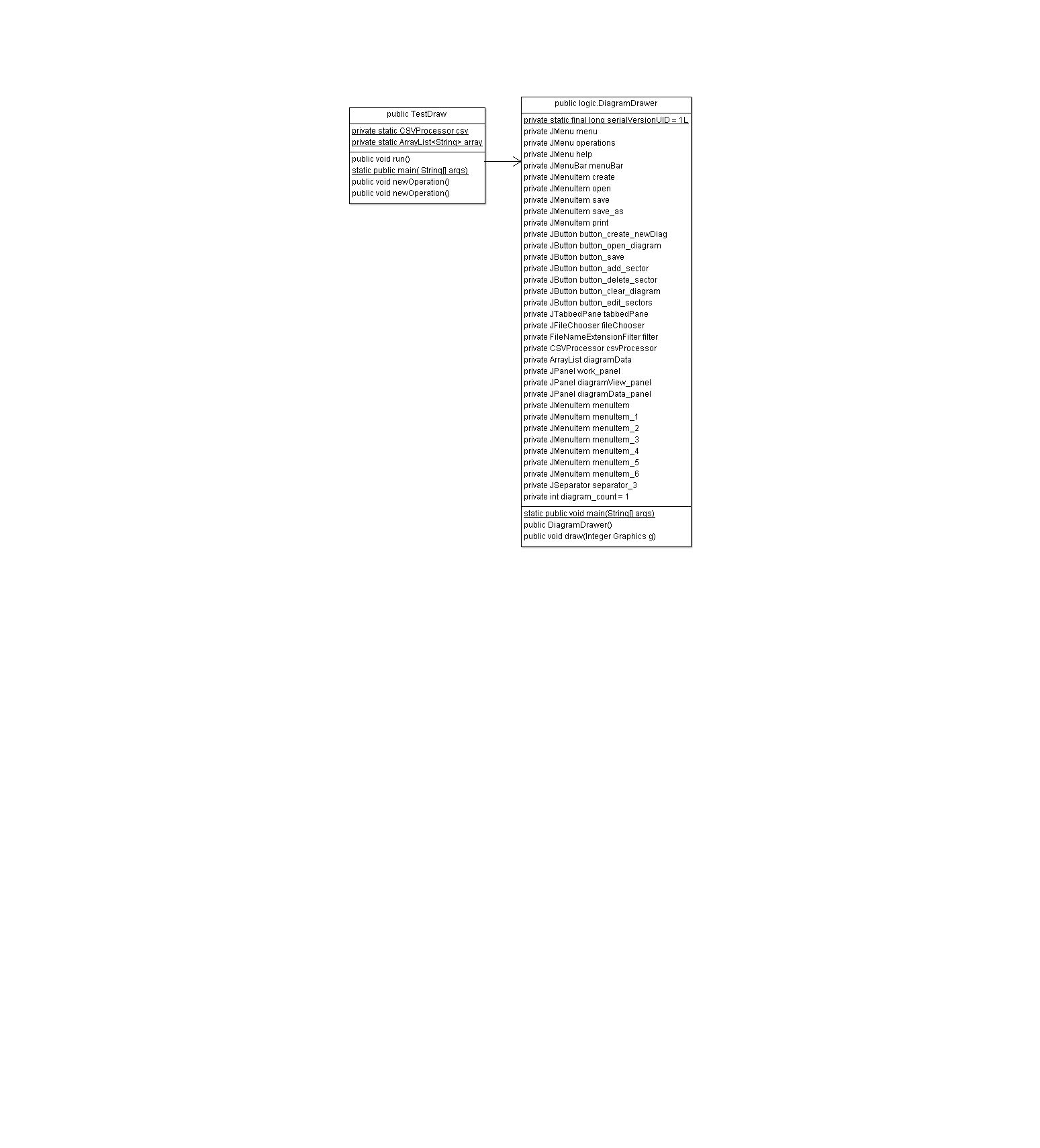
3. Ознайомитись з моделлю малювання в Java. Вільно володіти призначенням методів paint(),update() та repaint().

4. Розробити клас DiagramDrawer для відображення табличних даних у вигляді діаграми (згідно варіанту). Клас має містити метод draw(Graphics g) який здійснює малювання в графічному контексті g.

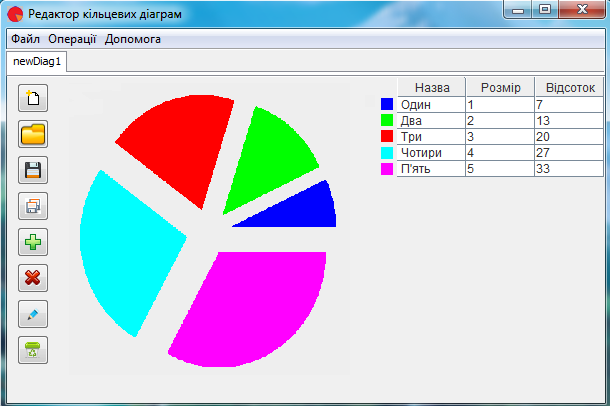
5. Створити тестовий клас TestDraw. В методі main():

* завантажити табличні дані з файлу і перевірити їх коректність (ЛР 2-3);
* створити екземпляр класу DiagramDrawer і ініціювати його завантаженними даними;
* створити екземпляр класу Frame, в якому екземпляр DiagramDrawer буде використовуватись в якості стратегії для малювання.

***Діаграма класів***



***Ілюстрація роботи програми***



***Лістинг коду***

**package** lab4;

**import java.awt.BorderLayout;**

**import java.awt.Color;**

**import java.awt.Graphics;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import javax.swing.JButton;**

**import javax.swing.JFrame;**

**import javax.swing.JLabel;**

**import javax.swing.JOptionPane;**

**import javax.swing.JTextField;**

**public class DiagramDrawer extends JFrame**

**{**

**private int CPU;**

**private int Mother;**

**private int OM;**

**private int Hard;**

**private int VGA;**

**private DiagramPanel diagrama = new DiagramPanel();**

**public Diagram()**

**{**

**setTitle("Diagram");**

**setBounds(400,100,600,600);**

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

**}**

**public void construct()**

**{**

**int sum = CPU + Mother + OM + Hard + VGA;**

**int i1 = (int)Math.round(CPU\*100./sum);**

**int i2 = (int)Math.round(Mother\*100./sum);**

**int i3 = (int)Math.round(OM\*100./sum);**

**int i4 = (int)Math.round(Hard\*100./sum);**

**int i5 = (int)Math.round(VGA\*100./sum);**

**diagrama.set(i1, i2, i3, i4, i5);**

**JLabel label = new JLabel("Процесор - " + i1 + "%");**

**JLabel label2 = new JLabel("Материнська плата - " + i2 + "%");**

**JLabel label3 = new JLabel("Оперативна пам'ять - " + i3 + "%");**

**JLabel label4 = new JLabel("Жорсткий диск - " + i4 + "%");**

**JLabel label5 = new JLabel("Відео карта - " + i5 + "%");**

**add(label);**

**label.setBounds(50, 20, 160, 26);**

**add(label2);**

**label2.setBounds(50, 46, 160, 26);**

**add(label3);**

**label3.setBounds(50, 72, 160, 26);**

**add(label4);**

**label4.setBounds(50, 97, 160, 26);**

**add(label5);**

**label5.setBounds(50, 124, 160, 26);**

**setLayout(new BorderLayout());**

**add(diagrama);**

**}**

**public void setCPU(String str)**

**{**

**CPU = Integer.parseInt(str);**

**}**

**public void setMother(String str)**

**{**

**Mother = Integer.parseInt(str);**

**}**

**public void setOM(String str)**

**{**

**OM = Integer.parseInt(str);**

**}**

**public void setHard(String str)**

**{**

**Hard = Integer.parseInt(str);**

**}**

**public void setVGA(String str)**

**{**

**VGA = Integer.parseInt(str);**

**}**

**}**  
**package** lab4;

**import** java.io.File;

**import** java.util.ArrayList;

**import** java.util.Scanner;

/\*\*

\* Клієнт.

\* **@author** Aruy

\*/

**public** **class** **TestDraw** {

**private** **static** CSVProcessor *csv*;

**private** **static** ArrayList<String> *array*;

/\*\*

\* Виводить на екран дані об'єкту типу ArrayList<String>

\* **@param** array об'єкт типу ArrayList<String>

\*/

**public** **static** **void** show(ArrayList<String> array, String s, String path) {

System.*out*.println (" " + s + " \"" + path.toString() + "\" :");

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

System.*out*.print(array.get(i));

System.*out*.println ();

}

/\*\*

\* Виводить на екран дані об'єкту типу String[][]

\* **@param** array об'єкт типу String[][]

\*/

**public** **static** **void** show\_mass(String[][] array) {

System.*out*.println (" Дані, що були перенесені з об'єкту типу ArrayList в String[][].");

**for** (**int** i = 0; i < array.length; i++) {

**for** (**int** j = 0; j < array[i].length; j++) {

System.*out*.print(array[i][j]);

}

}

}

**public** **static** **void** main(String[] args) **throws** Exception {

String ser\_path = "D:/serialized.dat";

File file = **null**;

File file\_ser = **new** File(ser\_path);

*csv* = **new** CSVProcessor(1024);

Scanner sc = **new** Scanner(System.*in*);

**byte** attempt = 3;

*array* = **null**;

Runnable r = **new** TestIO();

Thread t = **new** Thread(r);

t.join();

**if** (file\_ser.exists()) {

ArrayList<String> ser = *csv*.deserialization(file\_ser);

*show*(ser, "Дані десеріалізовані з файлу", ser\_path);

} **else** {

**do** {

**try** {

System.*out*.println (" Введіть шлях до CSV-файлу, звідки будуть зчитуватись дані.");

file = **new** File(sc.nextLine()/\*"D:/in.csv"\*/); // D:/in.csv

*array* = *csv*.nio\_read\_CSV(file);

} **catch** (CSVParseException e) {

attempt--;

**if** (attempt == 0) {

System.*out*.print ("У вас закінчились спроби!\n");

**throw** **new** Exception();

}

System.*out*.println (e);

System.*out*.println ("Помилка!!! Залишилось спроб: " + attempt + "\n");

}

} **while** (!file.exists()) ;

*csv*.serialization(*array*, file\_ser);

*show*(*array*, "Дані зчитані з файлу", file.getAbsolutePath());

t.start();

}

}

**public** **void** run() {

**try** {

*show\_mass*(*csv*.parse(*array*));

} **catch**(CSVParseException e) {

System.*out*.println (e);

}

}

}

***JavaDoc***

## C

[**CSVParseException**](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\CSVParseException.html) - Exception in [lab1](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\package-summary.html)

Клас виключення.

[**CSVParseException()**](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\CSVParseException.html#CSVParseException()) - Constructor for exception lab1.[CSVParseException](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\CSVParseException.html)

[**CSVProcessor**](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\CSVProcessor.html) - Class in [lab1](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\package-summary.html)

У класі CSVProcessor реалізувані методи завантаження/вивантаження CSV-файлу в/з агрегований об’єкт класу ArrayList, в якому знаходяться рядки CSV-файлу.

[**CSVProcessor(int)**](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\CSVProcessor.html#CSVProcessor(int)) - Constructor for class lab1.[CSVProcessor](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\CSVProcessor.html)

Конструктор

## D

[**deserialization(File)**](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\CSVProcessor.html#deserialization(java.io.File)) - Method in class lab1.[CSVProcessor](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\CSVProcessor.html)

Метод десеріалізує дані з файлу і повертає у вигляді об'єкту типу ArrayList

## G

[**getBuf\_size()**](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\CSVProcessor.html#getBuf_size()) - Static method in class lab1.[CSVProcessor](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\CSVProcessor.html)

## L

[lab1](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\package-summary.html) - package lab1

## M

[**main(String[])**](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\TestIO.html#main(java.lang.String[])) - Static method in class lab1.[TestIO](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\TestIO.html)

## N

[**nio\_read\_CSV(File)**](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\CSVProcessor.html#nio_read_CSV(java.io.File)) - Method in class lab1.[CSVProcessor](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\CSVProcessor.html)

Метод зчитує інформацію з CSV-файлу за допомогою потоків вводу пакету java.nio

[**nio\_write\_CSV(ArrayList<String>, File)**](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\CSVProcessor.html#nio_write_CSV(java.util.ArrayList, java.io.File)) - Method in class lab1.[CSVProcessor](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\CSVProcessor.html)

Метод записує дані з заданого масиву типу ArrayList в файл за допомогою потоків виводу пакету java.nio

## P

[**parse(ArrayList<String>)**](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\CSVProcessor.html#parse(java.util.ArrayList)) - Method in class lab1.[CSVProcessor](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\CSVProcessor.html)

Метод зчитує з об'єкту типу ArrayList дані і записує їх в об'єкт типу String[][].

## R

[**run()**](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\TestIO.html#run()) - Method in class lab1.[TestIO](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\TestIO.html)

## S

[**serialization(ArrayList<String>, File)**](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\CSVProcessor.html#serialization(java.util.ArrayList, java.io.File)) - Method in class lab1.[CSVProcessor](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\CSVProcessor.html)

Метод серіалізує дані об'єкту array у файл

[**setBuf\_size(int)**](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\CSVProcessor.html#setBuf_size(int)) - Static method in class lab1.[CSVProcessor](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\CSVProcessor.html)

[**show(ArrayList<String>, String, String)**](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\TestIO.html#show(java.util.ArrayList, java.lang.String, java.lang.String)) - Static method in class lab1.[TestIO](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\TestIO.html)

Виводить на екран дані об'єкту типу ArrayList

[**show\_mass(String[][])**](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\TestIO.html#show_mass(java.lang.String[][])) - Static method in class lab1.[TestIO](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\TestIO.html)

Виводить на екран дані об'єкту типу String[][]

## T

[**TestIO**](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\TestIO.html) - Class in [lab1](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\package-summary.html)

Клієнт.

[**TestIO()**](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\TestIO.html#TestIO()) - Constructor for class lab1.[TestIO](file:///D:\--==--\Java\%D0%9B%D0%B0%D0%B1%D0%B8\2\JavaDoc\lab1\TestIO.html)