

Лабораторні до дисципліни  
«Дослідження комп'ютерних систем штучного інтелекту»

У поясненнях даються можливі (не обов'язкові) варіанти розв'язків.

## Лабораторна №1

**Завдання:** Підготовка операційного середовища

Тема: конвертація повноколірних зображень.

Мета: метод/алгоритм конвертації файлів \*.bmp файлу (\*.jpg; \*.png; \*.gif) у файл \*.txt.

### Теоретичні відомості.

BMP (Від англ. Bitmap Picture) – Формат зберігання растрових зображень, розроблений компанією Microsoft.

З форматом BMP працює величезна кількість програм, так як його підтримка інтегрована в операційну систему Windows. Файли формату BMP можуть мати розширення \*.bmp, \*.dib і \*.rle. Крім того, дані цього формату включаються в двійкові файли ресурсів RES і в PE-файли.

Глибина кольору в даному форматі може бути 1, 4, 8, 16, 24, 32, 48 біт на піксель. При цьому для глибини кольору менше 16 біт використовується палітра з повноколірними компонентами глибиною 24 біта.

У форматі BMP зображення можуть зберігатися як є або ж із застосуванням деяких поширених алгоритмів стиснення. Зокрема, формат BMP підтримує RLE – стиснення без втрати якості, а сучасні операційні системи та програмне забезпечення дозволяють використовувати JPEG і PNG (ці формати вбудовуються в BMP як у контейнер).

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1m} \\ a_{21} & \cdot & & & \cdot \\ \cdot & & \cdot & & \cdot \\ \cdot & & & \cdot & \cdot \\ a_{n1} & & & & a_{nm} \end{pmatrix}$$

У результаті (виконання роботи) перетворення BMP файлу у txt файл (у двійковий формат). Текстовий файл повинен містити нулі і одинички (візуально видимі при відкритті у блокноті) по 8 у рядок, що відповідають RGB значенням кожного пікселя матриці  $A$ :

$$a_{11} \begin{cases} R | 00001111 \\ G | 01010101 \\ B | 10101010 \end{cases}$$

$$a_{12} \begin{cases} R | 01101111 \\ G | 01010001 \\ B | 10100011 \end{cases}$$

тощо. Тут  $a_{mn}$  – значення пікселя  $m$ -го стовпця в  $n$ -му рядку.

Файли \*.bmp та \*.txt потрібно записувати у наперед задану папку.

Для перевірки правильності виконаної процедури перетворення використовується обернене перетворення файлу \*.txt в \*.bmp.

### **Рішення (необов'язкові):**

Для середовища Delphi використовується get pixel, put pixel

В середовищі C++ використовується (triple)

В середовищі Java .

В середовищі C# .NET для вирішення даної задачі можна використовувати технологію *Windows form*.

Для роботи із зображенням використовуємо бібліотечний клас Bitmap та його методи GetPixel та SetPixel.

GetPixel(x, y) – отримує значення кольору пікселя з координатами x та y. Повертає об'єкт класу Color. Достаючи до полів R, G, B даного об'єкту отримуємо відповідні значення кольору даного пікселя.

SetPixel(x, y, color) – дозволяє встановити значення пікселя з координатами x та y кольором color.

Запис та зчитування з файлу здійснюється об'єктами класів StreamReader та StreamWriter.

### **Звіт.**

1. Огляд структури файлу BMP
2. е-програма (перетворення \*.bmp у \*.txt) засобами об'єктно-орієнтованого програмування;
3. пояснення роботи функцій програми.
4. Література, джерела, якими користувалися для реалізації.

Отримані навички: створення операційного середовища для роботи із зображеннями, наповнення БД,

### **Література**

1. Яглом А. М. Кореляційна теорія процесів з випадковими стаціонарними параметричними приростами // Математичний збірник. Т. 37. Вип.1. – С. 141-197. – 1955.
2. А.А. Свешніков Прикладні методи теорії випадкових функцій. – Гл.ред.фіз.-мат.літ., 1968.
3. С.І. Баскаков Радіо/технічні ланцюги і сигнали. – Вища школа. – 2000.

## Лабораторна №2

### Завдання:

1. Перехоплення програмою управління відеопотоком із відеокамери;
2. Перехоплення програмою управління відеопотоком із заданої відеокамери (з кількох можливих). (Зауважимо, що у notebook за замовчуванням вибирається інтегрована камера);;
3. Виділення наперед заданого кадру, наприклад, кожного 20-го або кожні 5мс., перетворення його в текстовий файл (лаб.№1) і його запис у задану папку;
4. Управління повинно здійснюватися з графічної оболонки;
5. Масштабування поля уваги (через форму за потрібними значеннями і за допомогою миші).
6. Завдання часу зйомки та запису (через форму поля), наприклад  $t=20\text{сек.}$ .

### Розв'язок (можливий) 1.1:

Якщо використовується C# .NET з технологією Windows Form.

Тоді для отримання відео/зображення з камери можна використати бібліотеку Aforge.

Спочатку отримуємо список камер. Даний список міститься в полі класу FilterCategory - VideoInputDevice. Після отримання списку камер викликаємо делегат, який на подію – створення нового кадру буде поміщати даний кадр в pictureBox, що забезпечить виведення зображень з камер.

\*\*\*

Мова вирішення: C# з бібліотекою MetriCam/

Завданням даної роботи є отримання зображень з камери та перетворення їх в текстовий формат.

Для отримання відео/зображення з камери використовуємо такі бібліотеки (Рисунок1)

- [ ] Bluetechnix
- [ ] Fotonic
- [ ] MesaCamera
- [ ] MetriCam
- [ ] MetriCamSDK
- [ ] Microsoft.CSharp
- [ ] O3DCamera
- [ ] PMDCamera
- [ ] PrimeSense
- [ ] SoftKinetic
- [ ] System
- [ ] System.Core
- [ ] System.Data
- [ ] System.Data.DataSetExtensions
- [ ] System.Drawing
- [ ] System.IO.Log
- [ ] System.Xml
- [ ] System.Xml.Linq
- [ ] WebCam
- [ ] XmlRpc

Рисунок. 1

### Розв'язок (можливий) 1.2.:

Якщо використовується Java, тоді можна використовувати бібліотеку webcam-capture. З цієї бібліотеки використовуються такі можливості, як пошук підключених до комп'ютера камер. Бібліотека дозволяє користувачу вибирати камеру для перехоплення зображення та відповідно перехоплювати саме зображення.

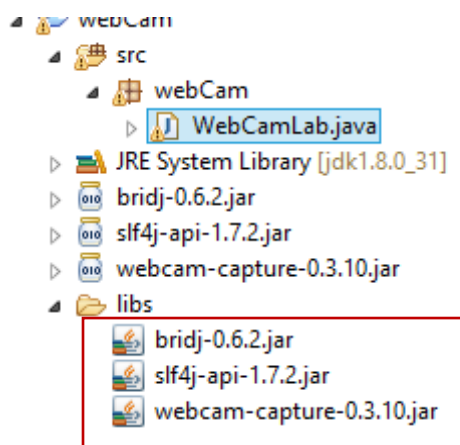
Для створення графічного інтерфейсу можна використовувати інструментарій *Swing*. Це набір бібліотек, який постачається з Java за замовчуванням.

Посилання на завантаження бібліотеки: <https://github.com/sarxos/webcam-capture/releases/download/webcam-capture-parent-0.3.10/webcam-capture-0.3.10-dist.zip>

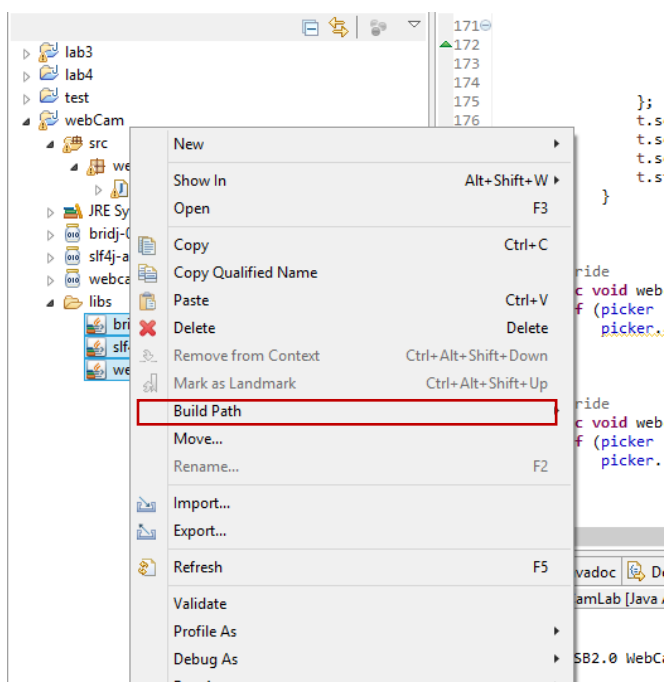
Підключення бібліотеки в проект eclipse:

1. В проекті створити папку `libs` і скопіювати в неї наступні файли з завантаженого архіву:

```
webcam-capture-0.3.10.jar
bridj-0.6.2.jar
slf4j-api-1.7.2.jar
```



2. Виділити дані файли, натиснути правою кнопкою миші і вибрати «Build Path» >> «Add to Build Path»



Зразок коду імпорту бібліотек:

```
package webCam;
```

```
import java.awt.BorderLayout;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;
import java.lang.Thread.UncaughtExceptionHandler;
```

```
import javax.swing.JFrame;  
import javax.swing.SwingUtilities;  
  
import com.github.sarxos.webcam.Webcam;  
import com.github.sarxos.webcam.WebcamDiscoveryEvent;  
import com.github.sarxos.webcam.WebcamDiscoveryListener;  
import com.github.sarxos.webcam.WebcamEvent;  
import com.github.sarxos.webcam.WebcamListener;  
import com.github.sarxos.webcam.WebcamPanel;  
import com.github.sarxos.webcam.WebcamPicker;  
import com.github.sarxos.webcam.WebcamResolution;
```

Документація до бібліотеки знаходиться в завантаженому архіві в папці docs.

### **Звіт.**

1. Огляд структури програми
2. е-програма;
3. пояснення роботи функцій програми.
4. Література, джерела, якими користувалися для реалізації; використані мова, методи, бібліотеки.

Отримані навички: створення операційного середовища для роботи із зображеннями, наповнення БД,

### **Джерела**

1. <http://www.webcam-capture.sarxos.pl/>
- 2.