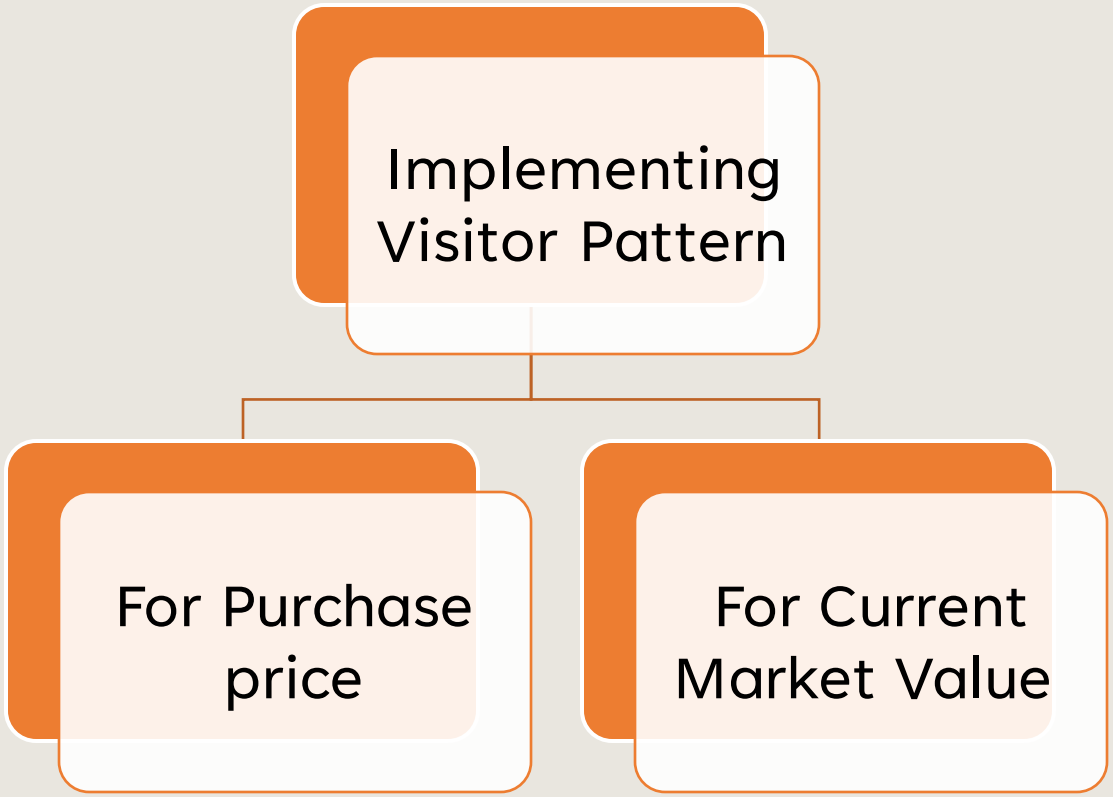
SOFTWARE ENGINEERING LAB

- Please demo your project status (Drafts) In Lab hours.
- Submission Date: Nov 30
- Per team only one submission is enough
- Please provide the individual contribution in the Declaration

VISITOR PATTERN



ADAPTER PATTERN

Add	Add a button “Launch Drone” to the drone actions area
Visit	Visit the items and item-containers placed on the farm at any location
Scan	Scan the Whole farm

SUBMISSION DOCUMENTS

- UML class diagrams PDF - how design patterns were implemented
- Readme file – how to run the project
- GitHub link
- Declaration
- References – if any

TELLO DRONE

User Guide 1: Tello SDK 2.0 User Guide: https://dl-cdn.ryzerobotics.com/downloads/Tello/Tello%20SDK%202.0%20User%20Guide.pdf

User Guide 2: Tello SDK Documentation: https://dl-cdn.ryzerobotics.com/downloads/tello/20180910/Tello%20SDK%20Documentation%20EN_1.3.pdf

For a sample Tello SDK code, you can access it here: https://uab.instructure.com/files/73597459/download?download_frd=1

Once you turn on the drone

Please connect your WIFI to Tello drone

If you check TelloDrone.java in the provided SDK you can see the drone is connected

Code

```
/*
 * Connection IP address.
 */
public static final String IP_ADDRESS = "192.168.10.1";

/*
 * Connection UDP Port.
 */
public static final Integer UDP_PORT = 8889, UDP_STATUS_PORT = 8890, UDP_VIDEO_PORT = 11111;

private int               battery, height, speed, time, temp, attitude[];
private int               missionPadId, missionPadxyz[], missionPadpry[];
private int               heading, headingZeroOffset = 9999, yawZeroOffset = 9999;
private double            barometer, tof, acceleration[], velocity[];
private String            sn, sdk;
private TelloConnection   telloConnection;
private TelloMode         telloMode;
private boolean           missionModeEnabled, flying;
private TelloModel        telloModel = TelloModel.EDU;

private static class SingletonHolder
{
    public static final TelloDrone INSTANCE = new TelloDrone();
}

/**
 * Get the global instance of TelloDrone.
 * @return Global TelloDrone instance.
 */
public static TelloDrone getInstance()
{
    return SingletonHolder.INSTANCE;
}
```

The TelloControl.java will initiate the drone connection and it consists of Tello command.

Code

```
public class TelloControl implements TelloControlInterface
{
    private final   Logger          logger = Logger.getLogger("Tello");
    private final   ConsoleHandler  handler = new ConsoleHandler();

    private TelloDrone          drone;

    private TelloCommunication  communication;


    private Thread              statusMonitorThread, keepAliveThread;

    // Private constructor, holder class and getInstance() implement this
    // class as a singleton.

    private TelloControl()
    {
        logger.setLevel(Level.OFF);
        handler.setLevel(Level.OFF);
        logger.addHandler(handler);
        logger.setUseParentHandlers(false);

        drone = TelloDrone.getInstance();

        communication = TelloCommunication.getInstance();


    }
```

TelloControl.java has the commands like Take off, land off, forward, backward. You can add more commands based on the instructions manual

Code

```
public void forward(Integer distance)
{
    TelloCommandInterface command = new ComplexTelloCommand(TelloCommandValues.FORWARD, distance.toString());
    communication.executeCommand(command);
}

@Override
public void backward(Integer distance)
{
    TelloCommandInterface command = new ComplexTelloCommand(TelloCommandValues.BACK, distance.toString());
    communication.executeCommand(command);
}

@Override
public void right(Integer distance)
{
    TelloCommandInterface command = new ComplexTelloCommand(TelloCommandValues.RIGHT, distance.toString());
    communication.executeCommand(command);
}

@Override
public void left(Integer distance)
{
    TelloCommandInterface command = new ComplexTelloCommand(TelloCommandValues.LEFT, distance.toString());
    communication.executeCommand(command);
}

@Override
public void rotateRight(Integer angle)
{
    TelloCommandInterface command = new ComplexTelloCommand(TelloCommandValues.CW, angle.toString());
    communication.executeCommand(command);
}
```

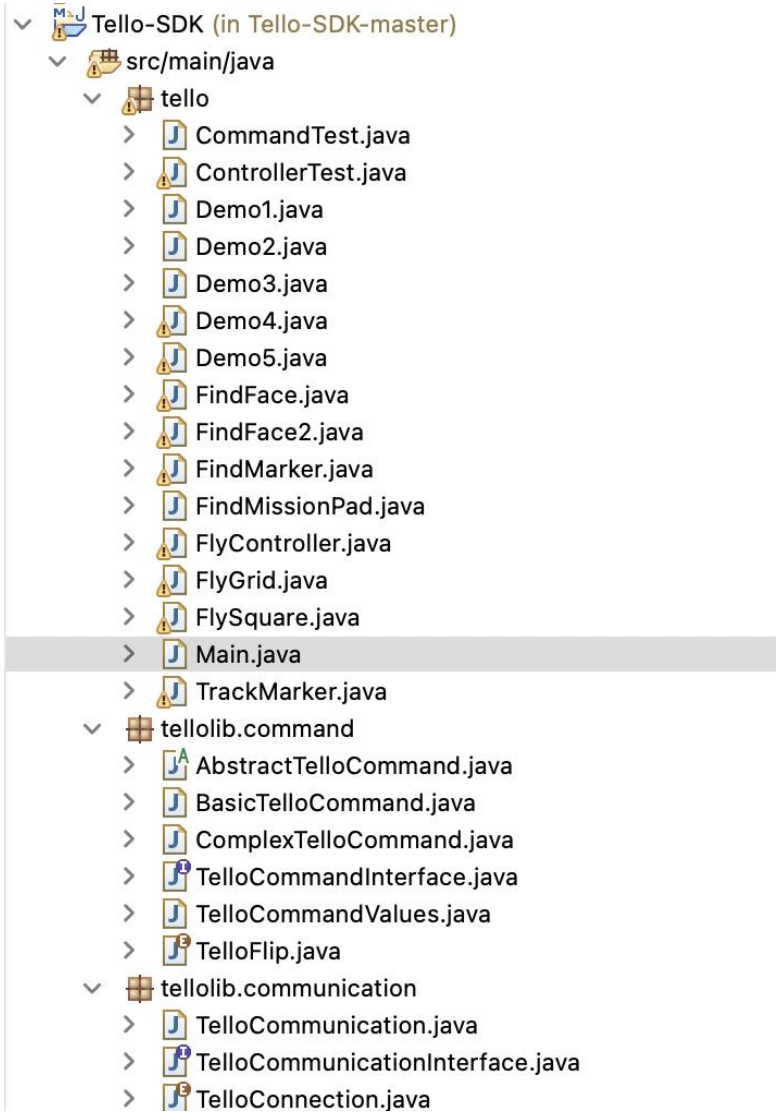
Different commands to execute the drone

```
telloControl.takeOff();

// Now we will execute a series of m
// Distances in centimeters.

telloControl.forward(50);

updateWindow();

telloControl.backward(50);

updateWindow();

telloControl.up(50);

updateWindow();

telloControl.down(50);

updateWindow();

telloControl.left(50);

updateWindow();

telloControl.right(50);

updateWindow();

telloControl.rotateLeft(90);

updateWindow();
```

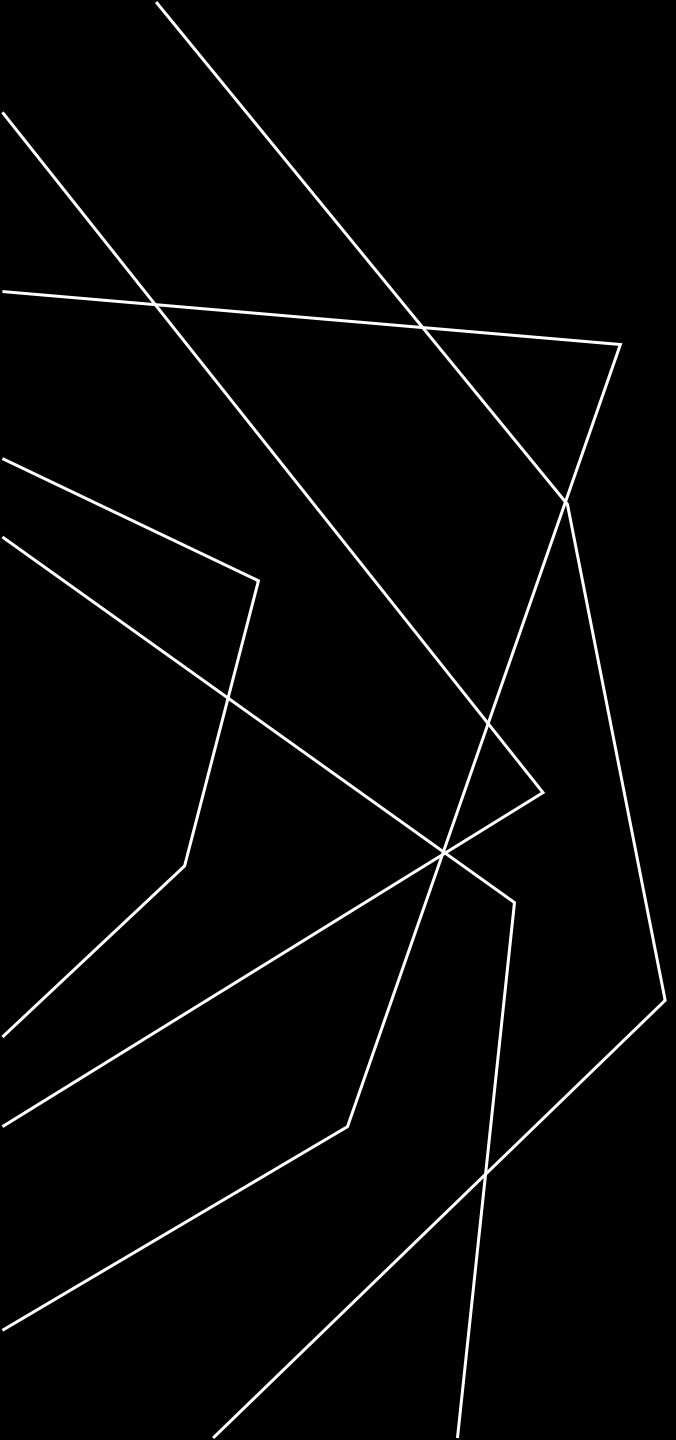
For the provided SDK there are couple of demos. You can play with it.

```
logger.info("start");

// Create an instance of the drone program

Demo1 demo = new Demo1();

Demo2 demo = new Demo2();

Demo3 demo = new Demo3();

Demo4 demo = new Demo4();

Demo5 demo = new Demo5();

FlySquare demo = new FlySquare();

FlyGrid demo = new FlyGrid();

FindMissionPad demo = new FindMissionPad()

FlyController demo = new FlyController();

FindMarker demo = new FindMarker();

TrackMarker demo = new TrackMarker();

FindFace demo = new FindFace();

FindFace2 demo = new FindFace2();
```

To run the SDK you can find the Main.Java inside the src/tello



Note: If you run into an error OpenCV library missing. Add the respective library to the project or else comment down the code which uses that library and check



THANK YOU