# Abstraction, Modularity and the Strategy Design Pattern

## Introduction

In this project we are commanding a drone to perform missions of user's choice.

## Aim

Our aim to is to implement Strategy Pattern which selects the mission on runtime and also introduce good Abstraction and Modularity.

## UML Diagrams

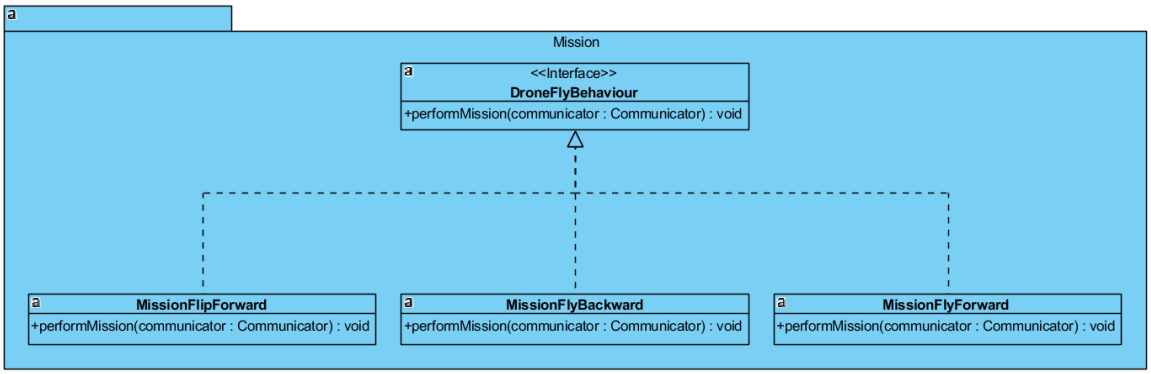


Fig. 1.1 UML Diagram of package Mission

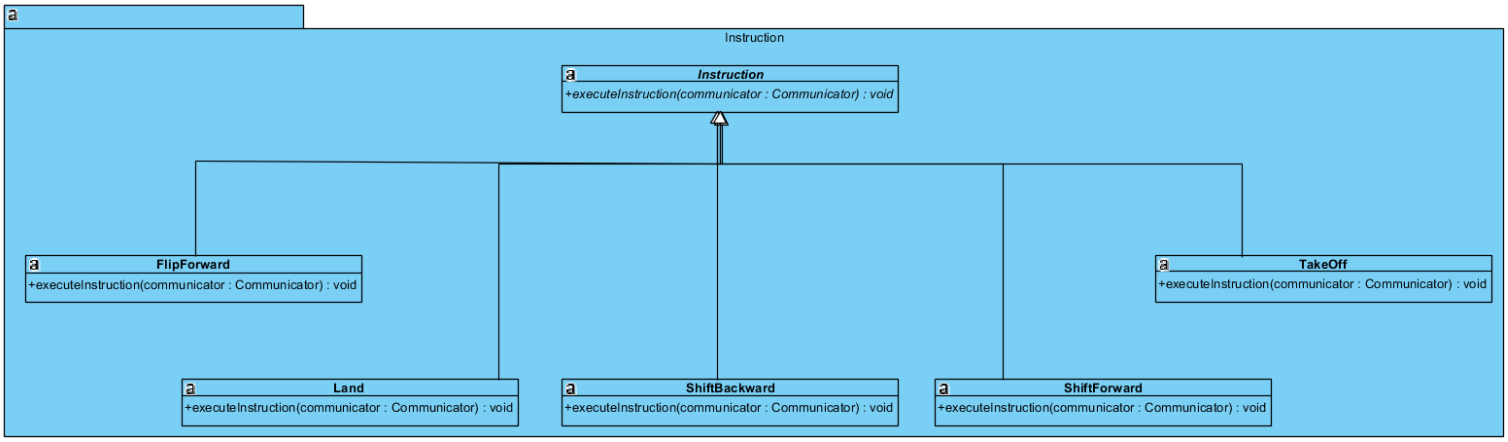


Fig 1.2 Diagram of package Instruction

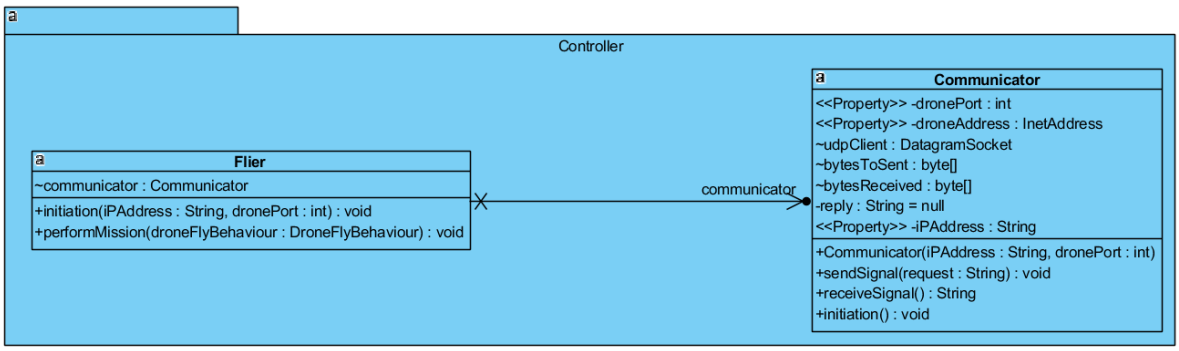


Fig 1.3 Diagram of Package Controller

## Insights Uncovered

-While trying to achieve the Strategy pattern I learned the importance of project structure which until now I have never focused on. Implementing the pattern actually gave me a generalized understanding of what kind of situation needs to follow strategy pattern. Designing a flow for the project before coding gave me a kick start and also was the key to implement good abstraction ad modularity.

-One very important fact that I have learned is regenerating the UML Diagram after completion of the project can give some astute observations about what changes can make the code more efficient. We can than make changes to the code and then again analyse the UML Diagram for better understanding. Conclusion to this point would be UML Diagram is not a one time thing but we should always find a way to improve its structure.

-I have learned and implemented that passing parameters via methods is a lot efficient than passing it through constructor(depends on the specific need though). These small things are the key to implement good abstraction and modularity.

-one of my bad practice that I tackled while doing this project is that we should always be open to new ideas and strategy rather than thinking in one direction. Because sometimes the only solution is to change the whole plan and get started again from a new point. I have wasted a lot of time trying to figure out things where all I needed was to think differently about a new solution.

-until now I have not been exposed to testing but now I am well aware of its importance which is no lesser than the other parts of the project.

-While doing the project I have increased my knowledge domain of java and every time we should try to figure out new methods and ways rather than coding just with the pre-existing knowledge.

-I have also figured out that the naming conventions are not to be taken lightly as it could make the code look messy and also it wont be readable for someone else. Also it is time wasting to debug a code which does not have proper names.

-Last point I learned is that we should always code in such a way that further addons and changes should not be a cumbersome task.