**SWE5202 Data Structures & Algorithms**

**Main Concepts**

* Inheritance
* Polymorphism

# Exercise 1: Airfield class without using polymorphism

Start Eclipse and create a project called **aircraft-3** and load all the classes infolder **aircraft-3** from Moodle. Look at the source code for the Airfield class you will see that there are 3 arraylists for the three types of plane. Examine the other methods and note the repetitive code needed to process 3 ArrayLists.

1. Add a main method to the Test Airfield class. Compile and run to get a console output.

Assume that we want to add a new type of plane to our game i.e. a mine layer which is used to drop mines into the sea to be exploded by passing ships.

We would have to -

1. create a new class called MineLayer which inherits from Aircraft.
2. add a field to store the MineLayer planes

private ArrayList<MineLayer> mlayers;

1. where necessary modify the methods to make use of this new field. The following table lists all the methods in Airfield, tick which ones need modifying.

|  |  |
| --- | --- |
| **Method** | **Change (y/n)** |
| public Airfield() | y |
| public void listAllPlanes() | y |
| public void addFighter(Fighter f) |  |
| public void addBomber(Bomber b) |  |
| public void addTorpedoPlane(TorpedoPlane t) |  |
| public void planesToAttack() | y |

1. We will need a new method to add a MineLayer write the code here.

|  |
| --- |
| Find code in zip file. |

# Exercise 2: Airfield class using polymorphism

Start Eclipse and create a project called **aircraft-4** and load all the classes in the folder **aircraft-4** from Moodle**.** Look at the source code for the Airfield class. Notice there is just one ArrayList, also the code for the methods listAllPlanes and planesToAttack is much simpler than in Airfield class from the project aircraft-3.

Compile the TestAirfield class, you should get the error message

**cannot find symbol – method useWeapons()**

This is because we have not added this method to any of our classes. We will correct the problem now.

# Exercise 3: Fixing the problem stage one

In the Aircraft class add the following method

/\*\*

\* This method has been added to enable polymorphism.

\*/

public void useWeapons(){

System.out.println("I have no weapons"); }

The TestAirfield class should now compile so create an object of type TestAirfield and then invoke the test\_001 method. It should list all the airplanes then tell them to use their weapons and then list them again. Examine the output did it work as expected? If not describe the problem here.

|  |
| --- |
| *The program runs but the aircrafts do not shoot.* |

# Exercise 4: Fixing the problem stage two

Add the following method to the Bomber class

/\*\*

* Making use of polymorphism.

\*/

public void useWeapons()

{ dropBombs();

}

now add the following code to the Fighter class

/\*\*

* Making use of polymorphism.

\*/ public void useWeapons()

{ fireGuns();

}

and now add the following code to the TorpedoPlane class

/\*\*

* Making use of polymorphism.

\*/ public void useWeapons()

{ dropTorp();

}

Create an object of type TestAirfield and then invoke the test\_001 method. It should list all the airplanes then tell them to use their weapons and then list them again. Examine the output. Did it work as expected?

*Yes the aircrafts are shooting properly and everything is working fine.*

# Exercise 5: Adding a MineLayer plane

Create a new class called MineLayer that inherits from Aircraft. Modify the TestAirfield class so that the method test\_001 also creates and add MineLayer planes to the Airfield. Did it work?

*It worked after I made all the changes in the tester class and made a new class called MineLayer.*

**NOTE under no circumstances should you modify the Airfield class, it is not necessary for this exercise.**