

## findLeastAndMostExpensiveAircraft()

### Method Code:

```
public static void findLeastAndMostExpensiveAircraft(Transportation[] list) {
    Aircraft leastExpensive = null;
    Aircraft mostExpensive = null;

    for (Transportation o : list) {
        if (list.length != 0 && list[0].getClass() != null && (o instanceof Aircraft)) {
            Aircraft aircraft = (Aircraft) o; //for using the method from the Aircraft class
            if (leastExpensive == null || aircraft.getPrice() < leastExpensive.getPrice()) {
                leastExpensive = aircraft;
            }

            if (mostExpensive == null || aircraft.getPrice() > mostExpensive.getPrice()) {
                mostExpensive = aircraft;
            }
        }
    }

    if (leastExpensive != null || mostExpensive != null) {
        if (list.length == 1) {
            System.out.println("Only one Aircraft found (The Least Expensive and " +
                               "the Most Expensive) is: " + list[0].toString());
        } else {
            System.out.println("The Least Expensive Aircraft:");
            System.out.println(leastExpensive);
            System.out.println("The Most Expensive Aircraft:");
            System.out.println(mostExpensive);
        }
    } else {
        System.out.println("Error, the list of aircraft is null, please enter valid list.");
    }
}
```

**Analysis:** The method finds the Least And Most Expensive Aircraft, if the list only has one Aircraft the Max and Min will be the same, if the array is null, it will display an error.

The parameter, list is as an array of the Transportation.

**Firstly**, creating two empty aircraft objects, one is for saving the least expensive aircraft from the array and the other is for saving the most expensive aircraft.

**Secondly**, in an enhanced for loop to find out them, and in the outer if conditions: make sure the array length is not 0 and there are some objects(transportsations) in the array, also

Transportation o should be under the Aircraft class (including WW2airplane). Then in the inner conditions, should make sure to cast the object into the aircraft and then use the method from aircraft class to compare their price by the getter method.

**Thirdly**, in the for loop, if the aircraft (object)'s price is smaller than leastExpensive (object) or the leastExpensive is null (for if only one aircraft from the array in case.)

**Finally**, after the end of the for loop, if at least one of the leastExpensive and mostExpensive is not null, there must be aircraft from the list array, then print them. Moreover, if the array's length is 1, which means there is only one aircraft, then this one is the least and most expensive. If not, then display the sentence to let the user know: that the list of aircraft is null.

## Output:

**//Creating 3 aircraft and 2 WW2 airplanes.**

```
Aircraft aircraft1 = new Aircraft(5, 7800);
Aircraft aircraft2 = new Aircraft(15, 9000);
Aircraft aircraft3 = new Aircraft();
WW2Airplane ww2Airplane1 = new WW2Airplane(4, 800, true);
WW2Airplane ww2Airplane2 = new WW2Airplane(5, 800, false);

//Create two arrays, allObjects have all 16 mixed objects,
allWithoutAircraft have 12 objects excluding aircraft1, aircraft2, aircraft3,
ww2Airplane1, ww2Airplane2,

Transportation[] allObjects = {wheeledTransportation1, wheeledTransportation2,
train1, train2, metro1, metro2, tram1, tram2, monowheel1, monowheel2,
aircraft1, aircraft2, aircraft3, ww2Airplane1, ww2Airplane2, ferry1, ferry2};
Transportation[] allWithoutAircraft = {wheeledTransportation1,
wheeledTransportation2, train1, train2, metro1, metro2, tram1, tram2,
monowheel1, monowheel2, ferry1, ferry2};

findLeastAndMostExpensiveAircraft(allObjects);

//output:
//The Least Expensive Aircraft:
//This WW2Airplane - 80000 price is 4.0 million, and it has a maximum elevation
of 800.0km And it has twin Engines
//Most Expensive Aircraft:
//This Aircraft - 1price is 15.0 million, and it has a maximum elevation of
9000.0km

findLeastAndMostExpensiveAircraft(allWithoutAircraft);

//output: Error, the list of aircraft is null, please enter valid list.
```

## copyTheObject()

### Method Code:

```
public static Transportation[] copyTheObjects(Transportation [] x){
    Transportation[] copyArr = new Transportation[x.length];
    for (int i = 0; i < x.length; i++) {
        //save the current object to find belongs to which class
        copyArr[i]=x[i].clone();
    }
    return copyArr;
}
```

### Analysis:

A static method for copying all the types of objects in this project

@parameter: x as the array of Transportation

@return type: copyArr as a new copied array of Transportation

**Firstly**, although Transportation is an abstract class in my project, but it can be created into an array for saving all objects from his child classes. Therefore, I created a new transportation array called copyArr with the same array length as parameter x which is the original array.

**Secondly**, in a for loop, save all the current objects into copyArr by using the abstract method clone().

The run-time system will use the clone method from each child class. (in each class the clone() method will return a new themselves object by its own copy constructor)

**Finally**, return copyArr as a new copied array of Transportation

### Output:

This is from the driver class's main method:

```
//Create an array, allObjects have all 16 mixed objects
Transportation[] allObjects = {wheeledTransportation1, wheeledTransportation2, train1, train2,
                               metro1, metro2, tram1, tram2, monowheel1, monowheel2,
                               aircraft1, aircraft2, ww2Airplane1, ww2Airplane2, ferry1, ferry2};
for (int i = 0; i < allObjects.length; i++) {
    System.out.println(i+1+" "+allObjects[i].toString());
}
System.out.println("-----");
System.out.println("These are copied objects:");
System.out.println();

Transportation[] copiedObjects = copyTheObjects(allObjects);
for (int i = 0; i < allObjects.length; i++) {
    System.out.println(i+1+" "+copiedObjects[i].toString());
}
```

Here is the output:

1. This WheeledTransportation - serial #0 - has 4 wheels, has a maximum speed of 120.0 km/hr.
2. This WheeledTransportation - serial #1 - has 2 wheels, has a maximum speed of 20.0 km/hr.
3. This Train - serial #10000 - has 48 wheels, has a maximum speed of 100.0 km/hr. It has 12 vehicles and its starting and destination stations are Toronto and Montreal
4. This Train - serial #10001 - has 36 wheels, has a maximum speed of 130.0 km/hr. It has 8 vehicles and its starting and destination stations are Montreal and Toronto
5. This Metro - serial #25000 - has 48 wheels, has a maximum speed of 70.0 km/hr. It has 6 vehicles and its starting and destination stations are Montreal West and Montreal East. The total stops are 18 stops.
6. This Metro - serial #25001 - has 48 wheels, has a maximum speed of 70.0 km/hr. It has 6 vehicles and its starting and destination stations are Montreal north and Montreal South. The total stops are 16 stops.
7. This Tram - serial #30000 - has 60 wheels, has a maximum speed of 110.0 km/hr. It has 12 vehicles and its starting and destination stations are Kingston and Montreal. The total stops are 6 stops. The year of creation is 1997
8. This Tram - serial #30001 - has 18 wheels, has a maximum speed of 100.0 km/hr. It has 10 vehicles and its starting and destination stations are Quebec City and Montreal. The total stops are 2 stops. The year of creation is 2000
9. This Monowheel - serial #55000 - has 1 wheel, has a maximum speed of 40.0 km/hr. The maximum weight is 100.0kg
10. This Monowheel - serial #55001 - has 1 wheel, has a maximum speed of 40.0 km/hr. The maximum weight is 100.0kg
11. This Aircraft - serial # 0 price is 5.0 million and it has a maximum elevation of 7800.0km
12. This Aircraft - serial # 1 price is 15.0 million and it has a maximum elevation of 9000.0km
13. This WW2Airplane - serial # 80000 price is 4.0 million and it has a maximum elevation of 800.0km it has twin Engines
14. This WW2Airplane - serial # 80001 price is 5.0 million and it has a maximum elevation of 7800.0km it has no twin Engines
15. This Ferry - serial #70000- has a maximum load of 300.0 kg, has a maximum speed of 80.0 km/hr.
16. This Ferry - serial #70001- has a maximum load of 500.0 kg, has a maximum speed of 70.0 km/hr.

-----  
These are copied objects:

1. This WheeledTransportation - serial #10 - has 4 wheels, has a maximum speed of 120.0 km/hr.
2. This WheeledTransportation - serial #11 - has 2 wheels, has a maximum speed of 20.0 km/hr.
3. This Train - serial #10006 - has 48 wheels, has a maximum speed of 100.0 km/hr. It has 12 vehicles and its starting and destination stations are Toronto and Montreal
4. This Train - serial #10007 - has 36 wheels, has a maximum speed of 130.0 km/hr. It has 8 vehicles and its starting and destination stations are Montreal and Toronto
5. This Metro - serial #25004 - has 48 wheels, has a maximum speed of 70.0 km/hr. It has 6 vehicles and its starting and destination stations are Montreal West and Montreal East. The total stops are 18 stops.
6. This Metro - serial #25005 - has 48 wheels, has a maximum speed of 70.0 km/hr. It has 6 vehicles and its starting and destination stations are Montreal north and Montreal South. The total stops are 16 stops.
7. This Tram - serial #30002 - has 60 wheels, has a maximum speed of 110.0 km/hr. It has 12 vehicles and its starting and destination stations are Kingston and Montreal. The total stops are 6 stops. The year of creation is 1997

8. This Tram - serial #30003 - has 18 wheels, has a maximum speed of 100.0 km/hr. It has 10 vehicles and its starting and destination stations are Quebec City and Montreal. The total stops are 2 stops. The year of creation is 2000
9. This Monowheel - serial #55002 - has 1 wheel, has a maximum speed of 40.0 km/hr. The maximum weight is 100.0kg
10. This Monowheel - serial #55003 - has 1 wheel, has a maximum speed of 40.0 km/hr. The maximum weight is 100.0kg
11. This Aircraft - serial # 4 price is 5.0 million and it has a maximum elevation of 7800.0km
12. This Aircraft - serial # 5 price is 15.0 million and it has a maximum elevation of 9000.0km
13. This WW2Airplane - serial # 80002 price is 4.0 million and it has a maximum elevation of 800.0km it has twin Engines
14. This WW2Airplane - serial # 80003 price is 5.0 million and it has a maximum elevation of 7800.0km it has no twin Engines
15. This Ferry - serial #70002- has a maximum load of 300.0 kg, has a maximum speed of 80.0 km/hr.
16. This Ferry - serial #70003- has a maximum load of 500.0 kg, has a maximum speed of 70.0 km/hr.