# **Intermediate Software Development – MiniProject 1**

## **Project Description:**

The goal of this project is to create object model for Solar System. An abstract super class called Planet and subclass called MilkyWayPlanet, with default constructor (no-argument) and full-argument overloaded constructors should be provided. Include an Interface class named Relatable which has two methods isMassSmaller() and isDiameterGreater() which compares the corresponding mass and diameter of any two planets. Ultimately, a driver class should be added in which instances are created for each planet, collects all planet objects into a Planet[] array named solarSystem, implement the polymorphic behavior, compare the mass of any two planets of our choice, compare the diameter of any two planets of our choice.

## **Installation, Compile and Run Time requirements:**

- 1. NetBeans IDE 7.2.1
- 2. Java 1.7.0\_11, Java Hotspot(TM) 64-Bit Server VM 23.6-b04
- 3. Windows 7 version 6.1 running on amd64

## Insights, Expected results, and Challenges:

### **Insights:**

This project helped me get hands on experience on the concepts of Inheritance, Polymorphism and Abstraction.

Inheritance was demonstrated by the subclass, which inherited the functionality from the super class and the interface.

The 'Planet' class demonstrated the characteristics of the 'MilkyWayPlanet' when it is instantiated with the subclass type. Thus demonstrating the concept of Polymorphism.

The concept of 'Abstraction' was demonstrated as follows:

The super class 'Planet' was made 'abstract', which means that the class cannot be instantiated and it is incomplete. However, the functionality of this class will be used by its subclass. I have learnt that by designing the abstract super class in a way that all the common fields are defined in it, it becomes easier to share the characteristics between the subclasses.

the packages used in this project are.	
□domain	
□driver	

The peelse are used in this project area

**domain** package consists of the following classes:

- 1. Planet
- 2. MilkyWayPlanet
- 3. Relatable

#### 1. Planet Superclass:

This class is an abstract super class. This is the parent class and it performs the following operations:

I.Import the required packages.

II.Declaration of variables.

III.Implementing the interface from the package Relatable.

IV.Providing No-arguments constructor with default values.

V. Providing Full-arguments constructor.

VI.Providing accessors and mutators for the variables declared.

VII.Providing toString() method.

### 2. MilkWayPlanet Subclass:

This class is a subclass which does the following operations:

I.Import the required package

II.Inherit the parent class i.e. the abstract super class 'Planet'

III.Create an instance of each planet.

#### 3. Relatable Interface class:

This is an interface which has the following methods:

I.boolean isMassSmaller(Object other ):

LO!	91	C	•

<del></del> -
☐ Initialize a boolean variable result to false.
□ Upcast the Object other to Planet super class
□ Compare the mass by retrieving the value from respective planets using
getmass() method.
☐ Change the value of boolean variable result if condition is true
□ Return the result value.

## II. boolean isDiameterGreater(Object other):

Logic:

- $\hfill\Box$  Initialize a boolean variable result to false.
- □ Upcast the Object other to Planet super class
- □ Compare the diameter by retrieving the value from respective planets using getmass() method.
- □ Change the value of boolean variable result if condition is true
- □ Return the result value.

driver package consists of the following class:

#### 1. Main Driver:

This is the main class from where the project execution starts and performs the following operations:

I.Planet's instance of MilkyWayPlanet is created.

II.Planet's data will be displayed.

III.All the Planet's objects are collected into an array of Planet called solarSystem.

IV.Implements polymorphism behavior where it displays Planet's data from solarSystem variable (Type Planet) using toString() method.

V. Invokes the isMassSmaller() comparison method for Mercury and Jupiter to compare the mass between two planets and displays which smaller. Invokes the

isDiameterGreater() comparison method for Mercury and Jupiter to compare the diameter between two planets and displays which is larger planet.

## **Expected Results:**

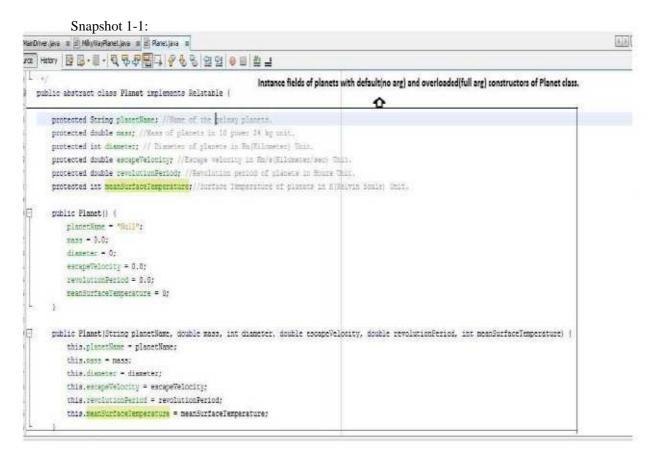
Expected Results	Test Result
All nine planets' data (created as instance of MilkyWayPlanet) should be displayed. [Not Mandatory as per the requirement]	Pass
2. All nine planets' data should use toString() method there by implementing polymorphism behavior(created by collecting all nine planets' object into an array of Planet called solarSystem).	Pass
3. Should display the result of comparison of any two planet's mass out of nine planets.	Pass
4. Should display the result of comparison of any two planet's diameter out of nine planets.	Pass

## Challenges Faced:

While coding the project, there were no major challenges as such. But did encounter minor hiccups while creating the test script that will write the output to a text file. Initially when I created the script file and executed, it created an empty text file. I tried several options to resolve it. Finally, I understood the problem was with jar file. So, I cleaned the project and build it which created a new jar file. Then when I executed the script file the output was successfully generated into text file as expected. Also, while populating data for instance fields of all planets' I noticed that the value of mass, revolution period etc varies from planet to planet in unit. For example Mercury revolution period was in days unit but Earth was in hours unit. Then I decided to use consistent unit for all the instance fields so that the result of mass and diameter comparisons would be accurate between any two planets.

## **Screenshots:**

## 1. Planet Superclass:



Snapshot 1-2:

```
Planet java 🔞
ree Hetory [10] 점·제·역 및 무선점 우선점 현점 (10) 점 21
       //Accessors and Mutators of instance fields
       public String getPlanetName() (
           return planetName;
       public void setPlanetName(String planetName) (
           this.planetName - planetName;
       public double getMass() (
           return mass;
       public void setMass(double mass) {
                                                            Accessors and Mutators of instance fields declared
           this.mass - mass;
                                                            in Planet class.
       public double getDiameter() (
           recurn diameter;
       public void setDiameter(int diameter) (
           this.diameter = diameter;
       public double getEscapeVelocity() (
          return escapeVelocity;
```

Snapshot 1-3: Planet iava 8 ore Hebry [2 등 - 후 · 및 등문문 다 상 등 등 전 이 표 호 교 ip public vois setEscapeVelocity |int escapeVelocity| ( this.esuspeVelocity = escapeVelocity; \$E public double getRevolutionPeriod() ( return revolutionFeriod; public void setRevolutionPeriod(double revolutionPeriod) ( this revolutionFewiod = revolutionPeriod; Accessors and Mutators of Planet superclass with 和日 public int getMeanSurfaceTemperature() { toString() method to format planets data. return meanJurGeseTemperature; t El public void setMeanSurfaceTemperature(int meanSurfaceTemperature) { this meanSurfaceTemperature - meanSurfaceTemperature; Soverride žΕ public String toString() ( return "Flamet" + "Planet Neme = " + planetHame + ", subs = " + mass + "a" + Marh.pov (10, 24) + "hi" + ", diameter = " + dispeter + "km" + ", emcapsVelocity = " + emcapeVelocity + "Wa/s" +
", revolutionPeriod = " + revolutionPeriod + "Houre" + ", meanSurfaceTemperature = " + meanSurfaceTemperature + "X" + ")';

## 2. MilkyWayPlanet subclass:

```
Snapshot 2-1:
ManDher java II il Milylla Planet java II il Planet java III
Source History 100 명 - 제 - 전 등 관 등 대 대 구 등 등 전 한 이 등 을 살 글
 4 - 4)
 5 package domain;
 7日 /**
 8
 3
     " Manthor MagnaVinay
                                                                        MilkyWayPlanet subclass with default no arg) and a full
10 - "
                                                                        arg, overloaded constructors and toString(| method.
11 public class MilkyWayPlanet extends Planet (
12
13
         //So argument Constructor
14 F
         public MilkyMayPlanet() |
15
            super():
16
17
18
         //full argument constructor
19 E
         public MilkyWayFlamet/String planet/Mane, double mass, int diameter, double escapeVelocity, double revolutionPeriod, int mannSurfaceTemperature) |
20
             super(planetName, mass, diameter, escapeVelocity, revolutionPeriod, meanSurfaceTemperature);
21
22
23
 8
         20verride
25 ⊞
         public String toString()
25
             return "Mikywayplanets" + super.toString() + "|";
20
28
25
         //Interface mathods ared
         //to compare mass of two planets
```

## Snapshot 2-2:

```
@Override
public boolean isMassSmaller(Object other) {
    boolean result = false;
    Planet p = null;
   if (other instanceof Planet) {
       p = (Planet) other;
   if (p != null) {
        if (this.getMass() < p.getMass()) {</pre>
            result = true;
                                                           Interface Methods is Mass Smaller (Object other)
    return result;
                                                           and isDiameterGreater(Object other) used in
                                                         Planet Class to compare mass and diameter of
//Interface methods used
                                                           any two planets
//to compare diameter of two planets
@Override
public boolean isDiameterGreater(Object other) {
   boolean result = false;
    Planet p = null;
   if (other instanceof Planet) {
       p = (Planet) other;
    if (p != null) {
       if (this.getDiameter() > p.getDiameter()) {
            result = true;
    return result;
```

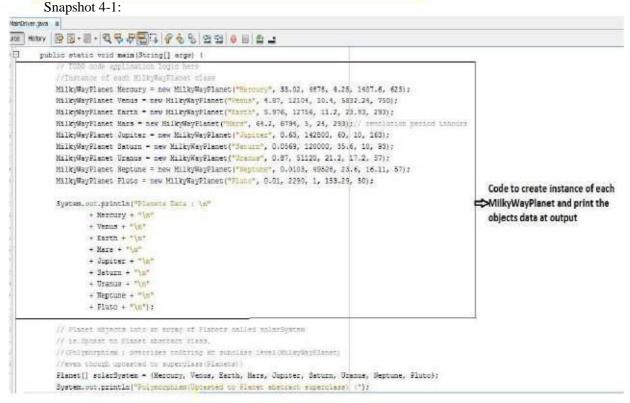
#### 3. Relatable Interface class:

### Snapshot 3-1

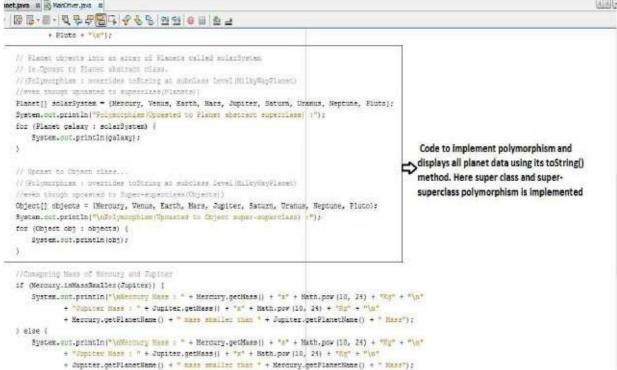
```
44 .
Relatable.java 🕸
ource History 🔯 🖫 - 🖫 - 💆 👺 👺 🖳 📮 🔗 😓 🖄 🖄 🔘 🔝 🎥 🚅
1 - /*
2 * T
3 * a
4 * /
    * To change this template, choose Tools | Templates
   * and open the template in the editor.
  package domain;
* @author MeghaVinay
   public interface Relatable {
                                              Interface methods isMassSmaller(Object other)
      boolean isMassSmaller(Object other);
      boolean isDiameterGreater(Object other);
⊕
.6 }
                                              Relatable interface class. These methods are
                                              further used in Planet class
```

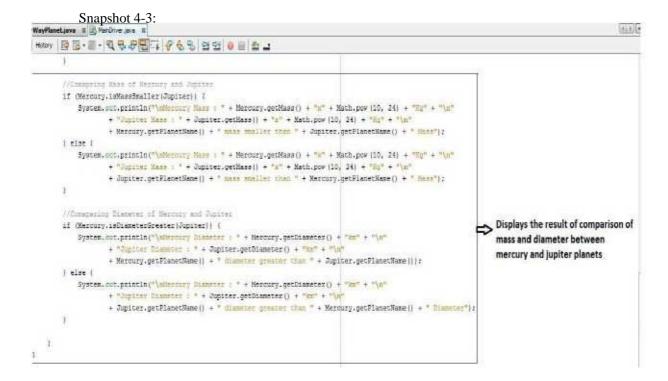
#### 4. MainDriver driver class:

Note: Currently the System.out.println("Planets.....); shown in Snapshot 4-1 is commented in the code as it wasn't the part of the requirement.



Snapshot 4-2:

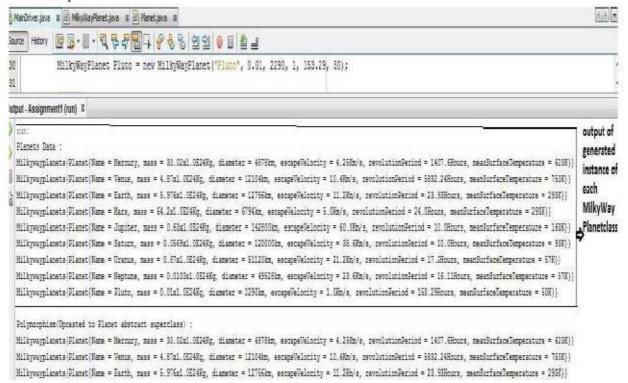




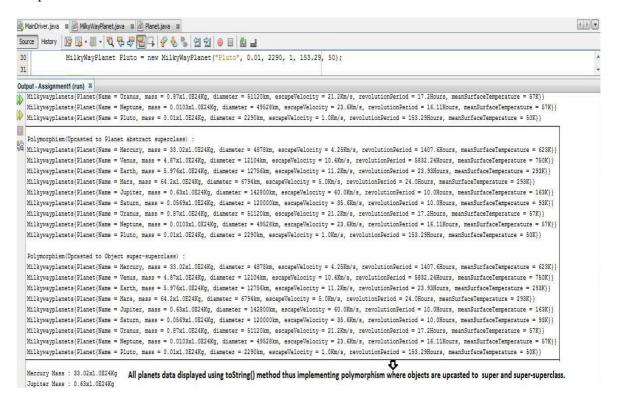
#### 5. Output:

Note: Initially I had coded to display the output for each instance of MilkyWayPlanet created but I have commented the code currently because it was not specified as the part of the comment. If needed, I can uncomment the code to display the result as shown in Snapshot 5-1

Snapshot 5-1:



#### Snapshot 5-2:



#### Snapshot 5-3:

