ITM 411 Mini Project 1: Read Me

1. Abstract

Aim: To create an object model for the Solar System

Objective: The objective of this mini project 1 was to help us understand the fundamental properties (APIE) of Object Oriented Programming (OOP)

Basic fundamentals of OOP:

- Abstraction: Abstraction is simplifying complex reality by modeling classes appropriate to the problem, and working at the most appropriate level of inheritance for a given aspect of the problem. It relates to generalization of the properties

- Polymorphism: It means having more than one form. Subclasses of a class can define their own unique behaviors and yet share some of the same functionality of the parent class

- Inheritance: It allows classes to inherit commonly used state and behavior from other classes. This relates to specialization of the properties(adding details to already existing state/behavior)

- Encapsulation: It facilitates the data composition with the methods (or other functions) operating on that data. It restricts access to some of the object's components. Often related to as 'Data Hiding'

2. System Requirement

OS: Windows 7

JDK: 1.7.0

IDE: NetBeans IDE 7.2.1

- Navigate to http://netbeans.org/ in order to download this version of NetBeans 7.2
- Click on Download

3. Architecture or System Flow

3.1 Creation of Java Application: Launch NetBeans and click on new Project to choose a new Java Application, name it as 'MiniProject1' and click on Finish

3.2 Creation of packages: Along with the default package, MiniProject1, create a new package 'domain' at the same level

3.3 Creation of classes/interfaces (Java files):

- Planet Class-

  This is an abstract method, which initializes all the field members(mass , diameter, revolution period and mean surface temperature).It implements the methods from the interface 'Relatable'

- MilkyWayPlanet Class-

  This class extends the Planet class. It inherits the protected class fields of the super class and has its own field 'planetName' to describe the name of a respective planet. It also provides an implementation for the 'escapeVelocity' abstract method

- Relatable Interface-

  This interface contains the declaration of two Boolean methods- isDiameterGreater() and isMassSmaller() whose implementation is provided in the 'Planet' class

All the above three classes are included in the package 'domain'

- MiniProject1 Class-

  This is the only class within the Miniproject1 package. It includes the main driver method which includes the following:

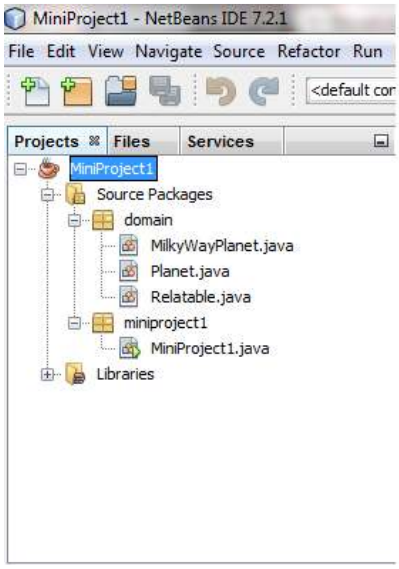
  1. Creation of 9 objects(mwp1…mwp9) of 'MilkyWayPlanet' class
  2. An array of Planet class called 'solarSystem' that collects all MilkyWayPlanet objects
  3. Calls to the two implemented methods for comparing the diameters and the masses of different planets

3.4 Script file: Created a 'script.bat' file located in the MiniProject1Docs folder. This file is responsible to convert the MiniProject1.jar file (This file is created after every successful build of the java code) into mp1out.txt file (located at the root level of the main project)

3.5 Javadoc API: This document depicts in detail in HTML format, all the comments written in the code.It can be created by right clicking on the java project and choosing the 'Generate Javadocs' option. This file folder gets automatically created inside the main project folder's 'dist' folder. It can be moved to a suitable location if needed

4. Screen Shots

4.1 Structure of the Java Project: The above screen capture shows the basic folder structure for the MiniProject1 JAVA application



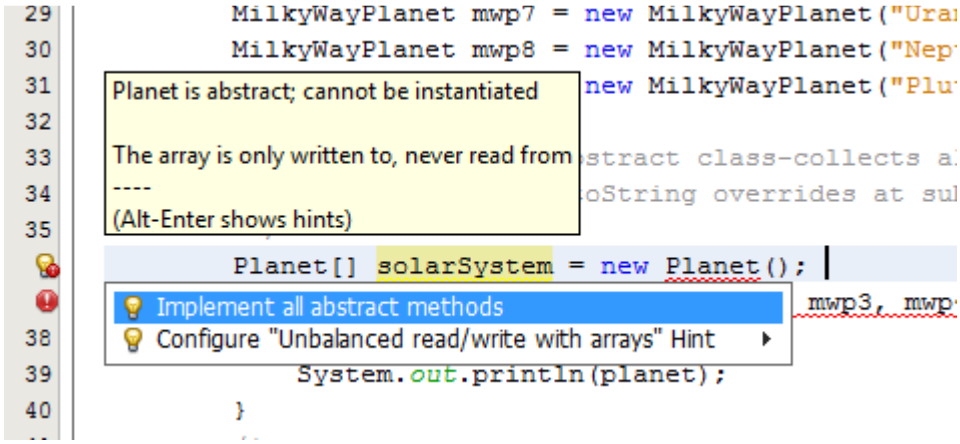
4.2 Output files:

```
run:
------------------------------------------------------------
MilkyWayPlanet: Name= Mercury
Facts:
Mass= 3.2999999999999996E23kg,
Diameter= 4870000.0m,
Revolution Period= 7603200.0sec,
Mean Surface Temperature= 800.0F
Escape Velocity= 4251.923106391907m/s
------------------------------------------------------------
MilkyWayPlanet: Name= Venus
Facts:
Mass= 4976.64kg,
Diameter= 1.21E7m,
Revolution Period= 1.944E7sec,
Mean Surface Temperature= 780.0F
Escape Velocity= 3.3125949878786886E-7m/s
------------------------------------------------------------
MilkyWayPlanet: Name= Earth
Facts:
Mass= 6113.28kg,
Diameter= 1.275E7m,
Revolution Period= 3.1536E7sec,
Mean Surface Temperature= 59.0F
Escape Velocity= 3.576637816523814E-7m/s
------------------------------------------------------------
MilkyWayPlanet: Name= Mars
```

```
MilkyWayPlanet: Name= Mars
Facts:
Mass= 5.2E25kg,
Diameter= 6790000.0m,
Revolution Period= 5.93568E7sec,
Mean Surface Temperature= -81.0F
Escape Velocity= 45202.212986867824m/s
-------------------------------------------
MilkyWayPlanet: Name= Jupiter
Facts:
Mass= 1.89E27kg,
Diameter= 1.42E8m,
Revolution Period= 1024704.0sec,
Mean Surface Temperature= 680.0F
Escape Velocity= 59590.85854839827m/s
-------------------------------------------
MilkyWayPlanet: Name= Saturn
Facts:
Mass= 5.68E26kg,
Diameter= 1.2E8m,
Revolution Period= 9.2905056E8sec,
Mean Surface Temperature= 332.3F
Escape Velocity= 35536.64775036235m/s
-------------------------------------------

MilkyWayPlanet: Name= Uranus
Facts:
Mass= 8.68E25kg,
Diameter= 5.11E7m,
Revolution Period= 2.64933936E9sec,
Mean Surface Temperature= -67.0F
Escape Velocity= 21288.365849635928m/s
-------------------------------------------
MilkyWayPlanet: Name= Neptune
Facts:
Mass= 1.0240000000000001E26kg,
Diameter= 4.95E7m,
Revolution Period= 14.052059999999999sec,
Mean Surface Temperature= -373.0F
Escape Velocity= 23493.098062317295m/s
-------------------------------------------
MilkyWayPlanet: Name= Pluto
Facts:
Mass= 1.3E22kg,
Diameter= 2360000.0m,
Revolution Period= 7.82E9sec,
Mean Surface Temperature= -380.0F
Escape Velocity= 1212.29576298408m/s

-------------------------------------------
--------------------------------------------------------------------------------

Jupiter's Diameter is greater than Mercury's Diameter!

--------------------------------------------------------------------------------

Earth's Mass is smaller than Jupiter's Mass!
BUILD SUCCESSFUL (total time: 0 seconds)
```

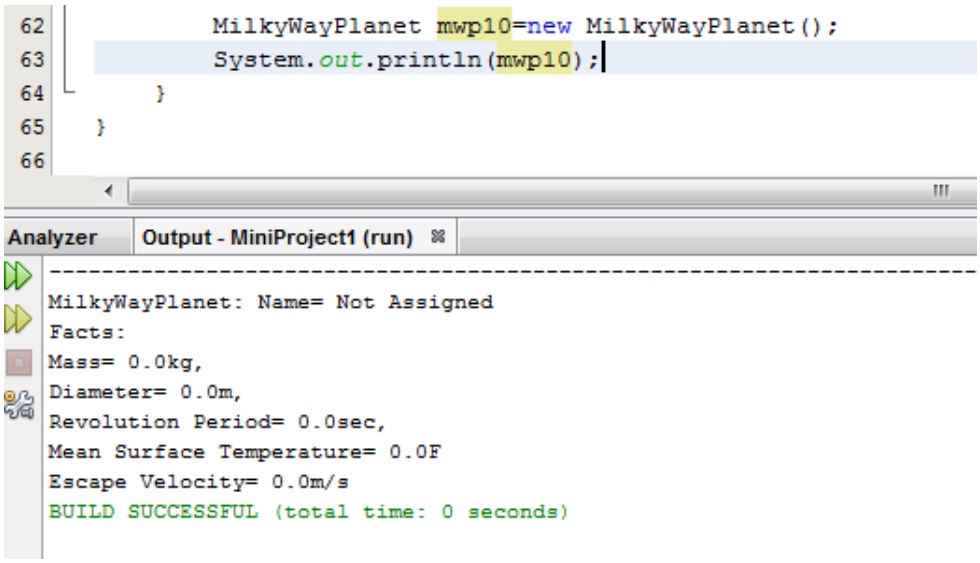
4.3 Test Cases:

- Test Case 1 – Negative



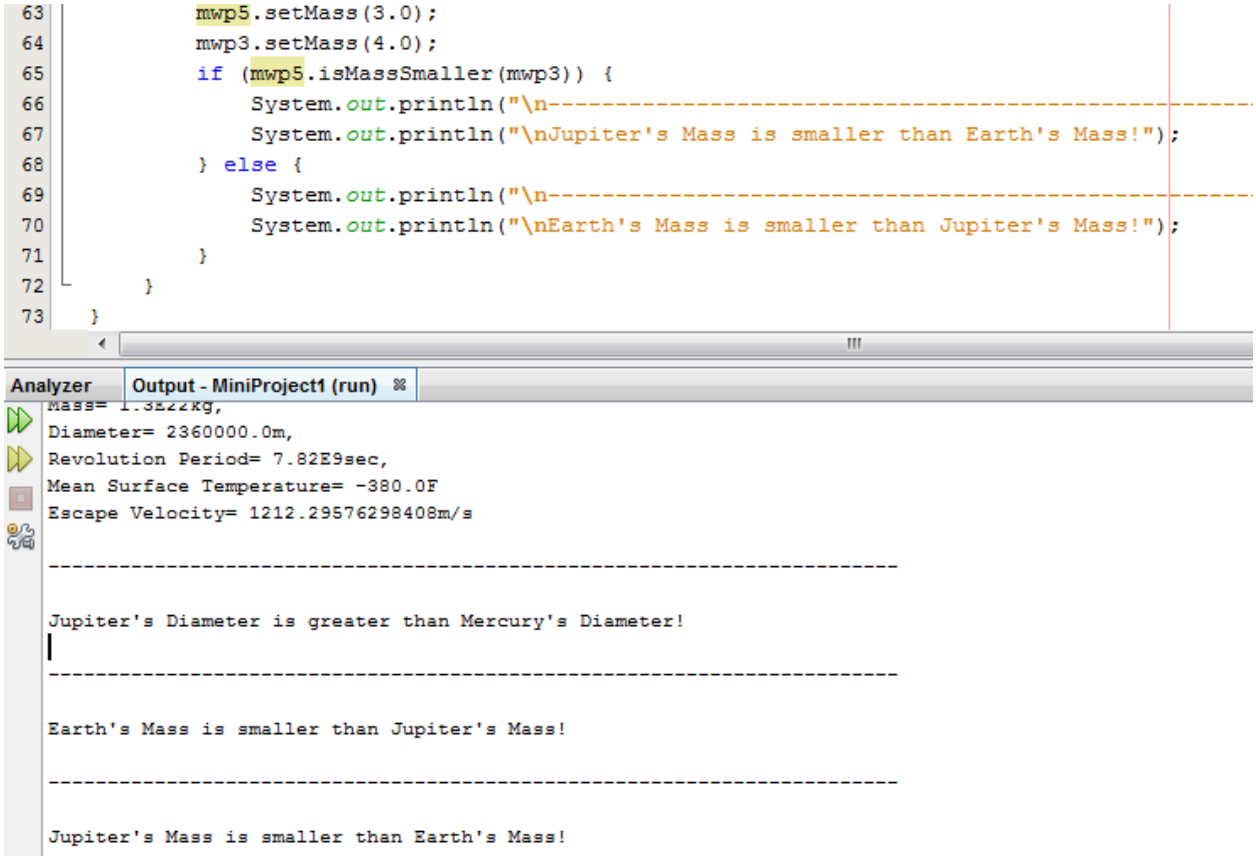
It shows that we cannot instantiate an abstract class.

- Test Case 2 – Positive



It created an additional instance (mwp10) of class 'MilkyWayPlanet' and printed it out.This shows that a call was given to a no args constructor of the Planet Class along with the subclass's no args constructor. (Depicts the inheritance property)

- Test Case 3 – Positive



```
63            mwp5.setMass(3.0);
64            mwp3.setMass(4.0);
65            if (mwp5.isMassSmaller(mwp3)) {
66                System.out.println("\n-----------------------------------------------------------
67                System.out.println("\nJupiter's Mass is smaller than Earth's Mass!");
68            } else {
69                System.out.println("\n-----------------------------------------------------------
70                System.out.println("\nEarth's Mass is smaller than Jupiter's Mass!");
71            }
72        }
73    }
```

```
Analyzer   Output - MiniProject1 (run)
Mass= 1.3822kg,
Diameter= 2360000.0m,
Revolution Period= 7.82E9sec,
Mean Surface Temperature= -380.0F
Escape Velocity= 1212.29576298408m/s

--------------------------------------------------------------------------

Jupiter's Diameter is greater than Mercury's Diameter!

--------------------------------------------------------------------------

Earth's Mass is smaller than Jupiter's Mass!

--------------------------------------------------------------------------

Jupiter's Mass is smaller than Earth's Mass!
```

This test case sets the value of ‘mass’ for the two objects mwp5 (Jupiter) and mwp3 (Earth) by accessing their mutator (setter) methods. The result can be verified by giving a call to ‘isMassSmaller()’

5. Conclusion

Thus, the code was implemented successfully. Please go through the mp1out.txt in order to view the output of the code.