Project Proposal

This software application will allow students to plan and design a complete Space Mission, build a Spaceship or multiple Spaceships and a Rover or multiple Rovers. The software application is a didactic aid for students.

They will need to make sure that the Spaceships and Rovers are cost effective, have an appropriate weight, velocity, and duration of mission so that it is successful.

The project will contain the following classes with the following data fields:

1. The Spaceship class:

* Serial Number (string with the format “XXX-123”)
* Height (double)
* Material (integer)
* Cost (double, will be calculated by software)
* Destination (integer)
* Weight (double, will be calculated by software)
* Radius entered by user
* Velocity (double, will be calculated by software)
* Duration of mission (double, will be calculated by software)
* Number of Spaceships (integer, static datafield and it will be calculated by the software)
* Success (boolean)

1. The Manned Spaceship class

* Number of people (integer)
* ArrayList of Rovers (Rover)
* Data Fields that are inherited from Spaceship class

1. The Unmanned Spaceship class

* ArrayList of Rovers (Rover)
* Data Fields that are inherited from Spaceship class

1. The Rover class

* Serial Number (string with the format “XXX-123”)
* Height (double)
* Material (integer)
* Cost (double)
* Weight (double)
* Function of rover(integer)
* Number of Rovers (integer, static datafield)

1. The Spaceship tester class

The Spaceship class will be an abstract class and the Manned Spaceship and the Unmanned Spaceship classes will extend it. The Manned and Unmanned Spaceship classes will be the aggregate classes for the Rover class. In the Aggregate classes, the Rovers will be stored using an ArrayList because it will allow the user to add as many Rovers as they want and it will expand automatically. To keep track of the number of Rovers created the class will contain a static data field, that will be accessed and mutated with static methods. Deep copies and deep returns will also be implemented in the Aggregate Classes to avoid any data leaks.

All object classes will contain a toString() for easy printing. The Rovers, Manned Spaceship and Unmanned Spaceship classes will contain a copy constructor for easy copying of objects and an equals() method, if two Rovers are equal the user will have an option to replace one of them with another Rover and if two Spaceships are equal then the user will be notified.

The tester will contain a polymorphic partially filled array of Spaceships that will have a size of twenty, to count of the number of Spaceships the class will have a static datafield, which will be accessed and mutated by static methods. To collect most of the information for the Spaceships the software will use a polymorphic loop.

To check the input of Height and Number of people the software will employ methods and do while loops. To make sure that serial numbers follow the format, each character will be checked using toCharArray, isDigit and isLetter.

All the user input and output will be stored in separate files (the software will also print the output) and possible exceptions will be handled with try and catch. The software will implement the CheckBox control to choose the Function of the Rover and the RadioButton controls to choose the Material and Destination. The software will also use GUIs to display an image of a Spaceship and a Rover that will have the message “ End of program” when the program ends.