

Programming Assignment 1

Introduction

This project is about using the principles of object-oriented programming we've learned to create a model of the solar system.

Classes

All classes should be part of a package named **solarsystem** and a package named **exceptions** (this will hold the custom exceptions we defined).

CelestialBody

In this class we have used the concept of inheritance as the star class and planet class are child classes so they can access the methods of parent class.

Must have the following variables:

- name (String) - private
- type (String) - private - holds type of CelestialBody (i.e. Star, Planet, Moon)
- Children (ArrayList<CelestialBody>) - private - holds orbiting children of CelestialBody

Must have the defined constructor:

- CelestialBody(String name, String type) - This will set the passed parameters to our defined variables (name and type).

Will implement the following methods:

- getName() - public - returns String - Returns the name of the CelestialBody
- getType() - public - returns String - Returns the type of the CelestialBody
- add(CelestialBody child) - public - void - Adds the passed CelestialBody to the children ArrayList if it is not already in the list, out prints a status that the CelestialBody was successfully added.
 - For example when Mercury is added to the Sun the method would out print "The **Planet Mercury** was added successfully to **Sun**".
 - If already in the list DuplicateCelestialBodyException is with a message. For example if Mercury is already in the list the exception message would be "The **Planet Mercury** is already in the collection"
 - Bold represents variables in the messages/outprints
- getChildren - public - returns ArrayList<CelestialBody> - returns orbiting children of CelestialBody

Star

Inherits from the class CelestialBody.

Must have the following variables:

- surfaceTemp (int) - private

Must have the defined constructor:

- Star(String name, int surfaceTemp) - This will send the passed parameter to the parent class, and hard code the type as Star. It will also set the passed surface temp of the star.

Will implement the following method:

- getSurfaceTemp() - public - returns int - Returns the surfaceTemp of the Star

IObit

This is an interface.

This will have one defined method:

- getOrbit() - void

Planet

Inherits from the class CelestialBody. Implements the Interface IObit.

Must have the following variables:

- orbits (CelestialBody) - private

Must have the defined constructor:

- Planet(String name, CelestialBody orbits) - This will send the passed parameter to the parent class, and hard code the type as Planet. The orbits variable will be set to the CelestialBody passed in if the CelestialBody has the type "Star" otherwise and InvalidCelestialBodyException will be thrown, with the message "A planet must orbit a star."

Will implement the following method:

- getOrbit() - public - void - Outprint what the Planet is orbiting. For example if earth.getOrbit() was called the outprint would be "**Earth** is orbiting the **Star Sun**"
 - Bold represents variables in the messages/outprints

Moon

Inherits from the class CelestialBody. Implements the Interface IOrbit.

Must have the following variables:

- orbits (CelestialBody) - private

Must have the defined constructor:

- Moon(String name, CelestialBody orbits) - This will send the passed parameter to the parent class, and hard code the type as Moon. The orbits variable will be set to the CelestialBody passed in if the CelestialBody has the type "Planet" otherwise and InvalidCelestialBodyException will be thrown, with the message "A moon must orbit a planet."

Will implement the following method:

- getOrbit() - public - void - Outprint what the Planet is orbiting. For example if moon.getOrbit() was called the outprint would be "**Moon** is orbiting the **Planet Earth**"
 - Bold represents variables in the messages/outprints

DuplicateCelestialBodyException

Will be in the **exceptions** package. This will extend Exception.

Must have the defined constructor:

- DuplicateCelestialBodyException(String message)

InvalidCelestialBodyException

Will be in the **exceptions** package. This will extend Exception.

Must have the defined constructor:

- InvalidCelestialBodyException(String message)

SolarSystem

The main class which will execute the program (i.e. the Solar System model), here we will define our Star and Planets.

In this class

- The main method must be implemented and handle the DuplicateCelestialBodyException (i.e. no throws declaration)
- 1 star (the Sun) must be defined (surface temp is ~5778k).

- 8 planets (Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, Neptune) must be defined and added to the Star Sun.
- The Moon must be defined and added to planet Earth.
- Phobos and Deimos must be defined and added to planet Mars.
- No other moons need to be defined.

Submission

After completing this programming assignment, zip up all the classes under your **src** folder (**solarsystem** and **exceptions** must be in the zip), the name of the zip file **MUST** be ProgrammingAssignment01_YY_XXXX.zip (YY = 01,02 or 03 and XXXX = student-id).