
Lab 18 - Multiple Inheritance

A true Object-Oriented Solar_System

AdaCore

The purpose of this exercise is to rewrite the previous exercise using OOP.

Remember that privacy is still important!

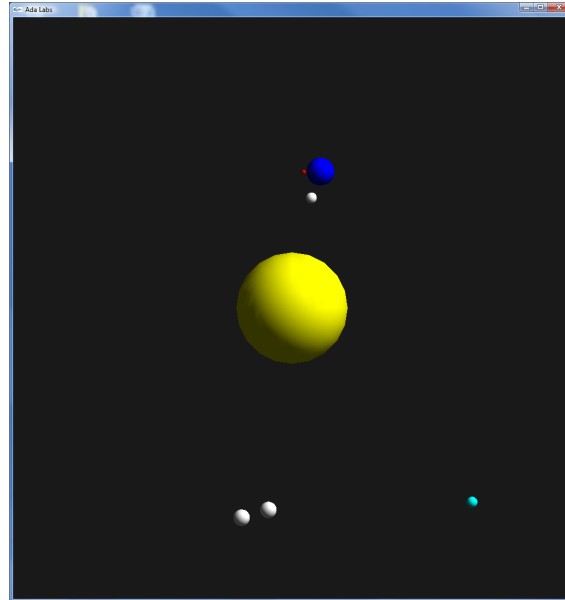


Figure 1: Expected result

Question 1

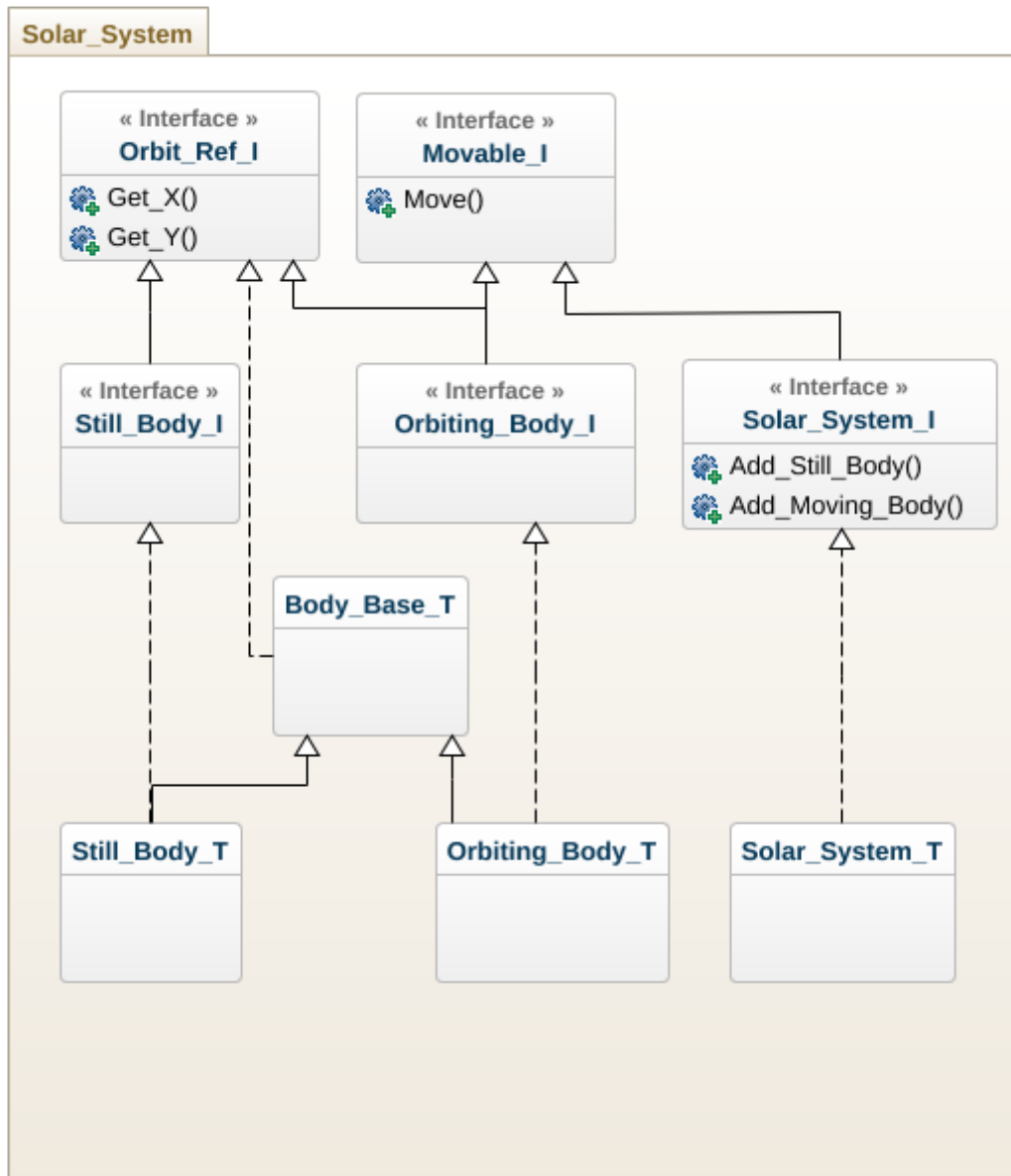


Figure 2: Class diagram for Q.1 and Q.2

Create a hierarchy of interfaces as follow

- `Orbit_Ref_I` as an interface implementing `Get_X` and `Get_Y` (can be used as an orbit refer-

ence)

- `Movable_I` as an interface implementing a `Move` procedure
- `Orbiting_Body_I` as an interface implementing `Orbit_Ref_I` and `Movable_I`
- `Still_Body_I` as an interface implementing `Orbit_Ref_I`
- `Solar_System_I` as an interface implementing `Add_Still_Body` and `Add_Moving_Body` procedures and `Movable_I`

Question 2

Create a hierarchy of tagged types as follow

- `Body_Base_T` to store a position (X, Y) and implementing `Orbit_Ref_I`
- `Orbiting_Body_T` as a concrete object extending
- `Body_Base_T` to store `Distance`, `Speed`, `Angle` and `Turns_Around` and implementing `Orbiting_Body_I`
- `Still_Body_T` as a concrete object extending `Body_Base_T` and implementing `Still_Body_I`
- `Solar_System_T` as a concrete object implementing `Solar_System_I` able to store a vector of moving bodies and a vector of still bodies

Question 3

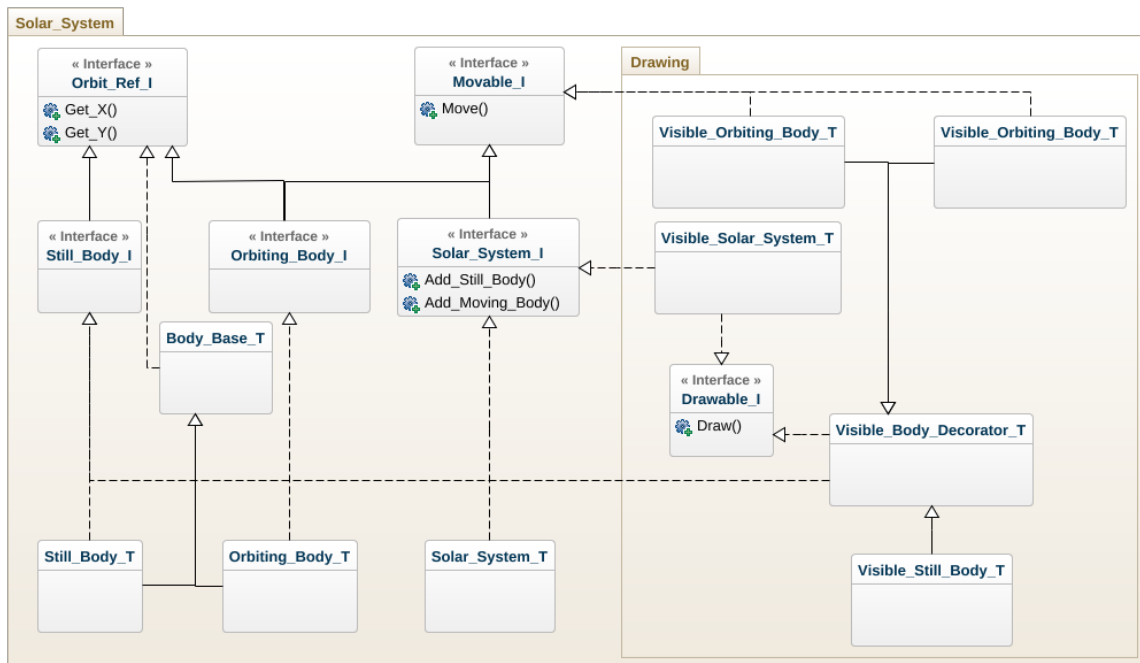


Figure 3: Class diagram for Q.3, Q.4 and Q.5

Add constructor primitives in order to create concrete types returning a pointer to the newly allocated object

- Create_Orbiting returning an access to Orbiting_Body_T
- Create_Still returning an access to Still_Body_T
- Create_Solar_System returning an access to Solar_System_T

Question 4

In a specific Graphics package extend the capabilities of our object using decorator design pattern.

- Create an interface Drawable_I implementing a Draw procedure
- Define Visible_Body_Decorator_T as an abstract type implementing Drawable_I and Still_Body_I. This type will store graphic information.
- Define Visible_Orbiting_Body_T extending Visible_Body_Decorator_T and implementing Movable_I

- Define `Visible_Still_Body_T` extending `Visible_Body_Decorator_T`
- Define `Visible_Solar_System_T` implementing `Drawable_I` and `Solar_System_I`

Question 5

- Add constructor `Create_Visible` to `Visible_Orbiting_Body_T`
- Add constructor `Create_Visible` to `Visible_Still_Body_T`
- Add constructor `Create_Visible` to `Visible_Solar_System_T`

Question 6

Make it work ! ;)