University of Strathclyde Glasgow

Python: Designing classes

These exercises assume that the “Python: Classes” exercises have been completed.

1. Write a `__repr__(self)` function for the `ElectricCar` class, which should return a string representation of the object. The function should allow a new object to be created by using `eval` on the returned string:

```
car = ElectricCar(10, 50, 20, 150, 1000)
new_car = eval(str(car))
```

2. Create a UML representation of the `ElectricCar` class, including the attributes, operations and visibility.
3. Write a `__repr__(self)` function for the `SolarArray` and `SolarPanel` class, which should return the string representation of the objects. The function should allow a new object to be created by using `eval` on the returned string.
4. Create a UML representation of the `SolarArray` and `SolarPanel` class, including the attributes, operations and visibility.
5. Add a `__repr__(self)` function to the `Customer` and `Purchase` classes to return the contents of the class as a string. The function should allow a new object to be created by using `eval` on the returned string
6. Create a UML representation of the `Customer` and `Purchase` class, including the attributes, operations and visibility.
7. Create three classes to match the UML diagram that is given in Figure 1.
   - The `Shape` constructor should assign values to `width`, `height` and `depth`.
   - `area_hd()` should return `height` $\times$ `depth`.
   - `area_wd()` should return `width` $\times$ `depth`.
   - `area_wh()` should return `width` $\times$ `height`.
   - The `area()` for `Rectangle` should return the value of `width` $\times$ `height`.
   - The `volume()` for `Rectangle` should return `0`.
   - The `area()` for `RectangularPrism` should return the value of:
     $2 \times ($ `width` $\times$ `height` + `width` $\times$ `depth` + `height` $\times$ `depth` $)$
   - The `volume()` for `RectangularPrism` should return the value of:
     `width` $\times$ `height` $\times$ `depth`
   - Add `__repr__(self)` functions for the three classes.
   - Create a program that uses the three classes.

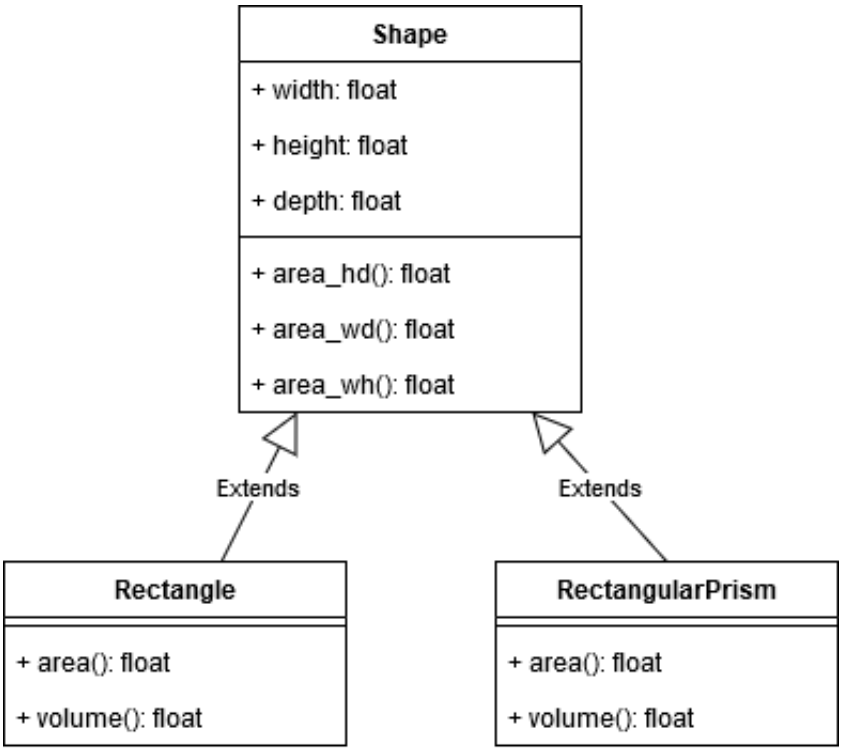

Figure 1: UML design of three classes.