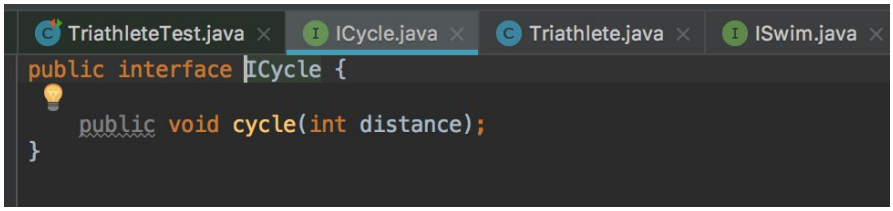
Polymorphism & Composition Homework - Quiz

Polymorphism

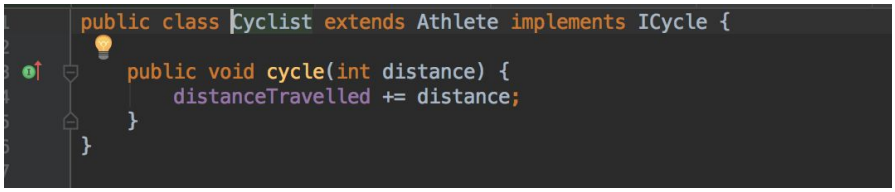
1. What does the *word* 'polymorphism' mean?
    - Something that can have many forms e.g. Sarah can be a student, a wife, a daughter, a swimmer etc…
    - A netball player can be a center-court player and a shooter (e.g. GA)
2. What does it mean when we apply polymorphism to OO design? Give a simple Java example.
    - When we can treat an instance of a class as if it is also another class at the same time
    - e.g a banana is a class and is also a fruit
    - E.g. cake, ice-cream and meringue are classes but are also a “pudding” class
3. What can we use to implement polymorphism in Java?
    - We can use an abstract class e.g. properties and methods are inherited but this means all the children of the parent class adopt these properties
    - Interface - description of methods that classes need to implement - like a contract - the class has to implement all the methods set out in the interface.
    - A class can implement more than one interface but can only extend one abstract class
4. How many 'forms' can an object take when using polymorphism?
    - As many as you like
5. Give an example of when you could use polymorphism.
    - In the Triathlete example, we wanted a class of athlete that would enable a triathlete, swimmer, runner and cyclist to inherit the athlete properties. However we didn’t want to write methods for swimming, cycling and running in the abstract class as they would only be all applicable to the triathlete - so by creating an interface for swimming, cycling and running

we could implement these in the appropriate classes for swimmer, cyclist and runner and implement them all in the triathlete class.

ICycle method:

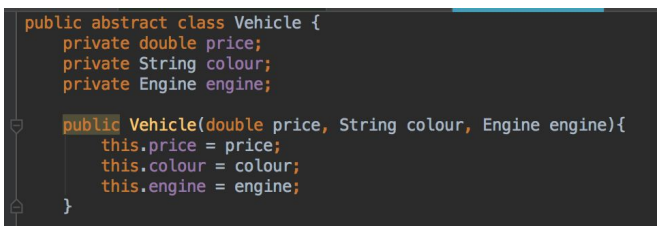


Implementing the interface:



Composition

1. What do we mean by 'composition' in reference to object-oriented programming?
    a. When an object is made up of other object(s) e.g. “Has a...”
    b. e.g. a house has a window, door, bedroom,
    c. A netball team is comprised of players, each with their own methods (e.g all players can run, but only certain players can shoot)
2. When would you use composition? Provide a simple example in Java.
    a. To make an object up from smaller things
    b. E.g a car is made up of an engine which has its own class with properties.



3. What is/are the advantage(s) of using composition?
    a. If you use inheritance, all of the classes that inherit from the superclass need to have the properties and methods of that parent class - by using composition you only are using the behaviours and properties that you need.
4. What happens to the behaviours when the object composed of them is destroyed?
    a. They are also destroyed - all objects that it is composed of are also destroyed.