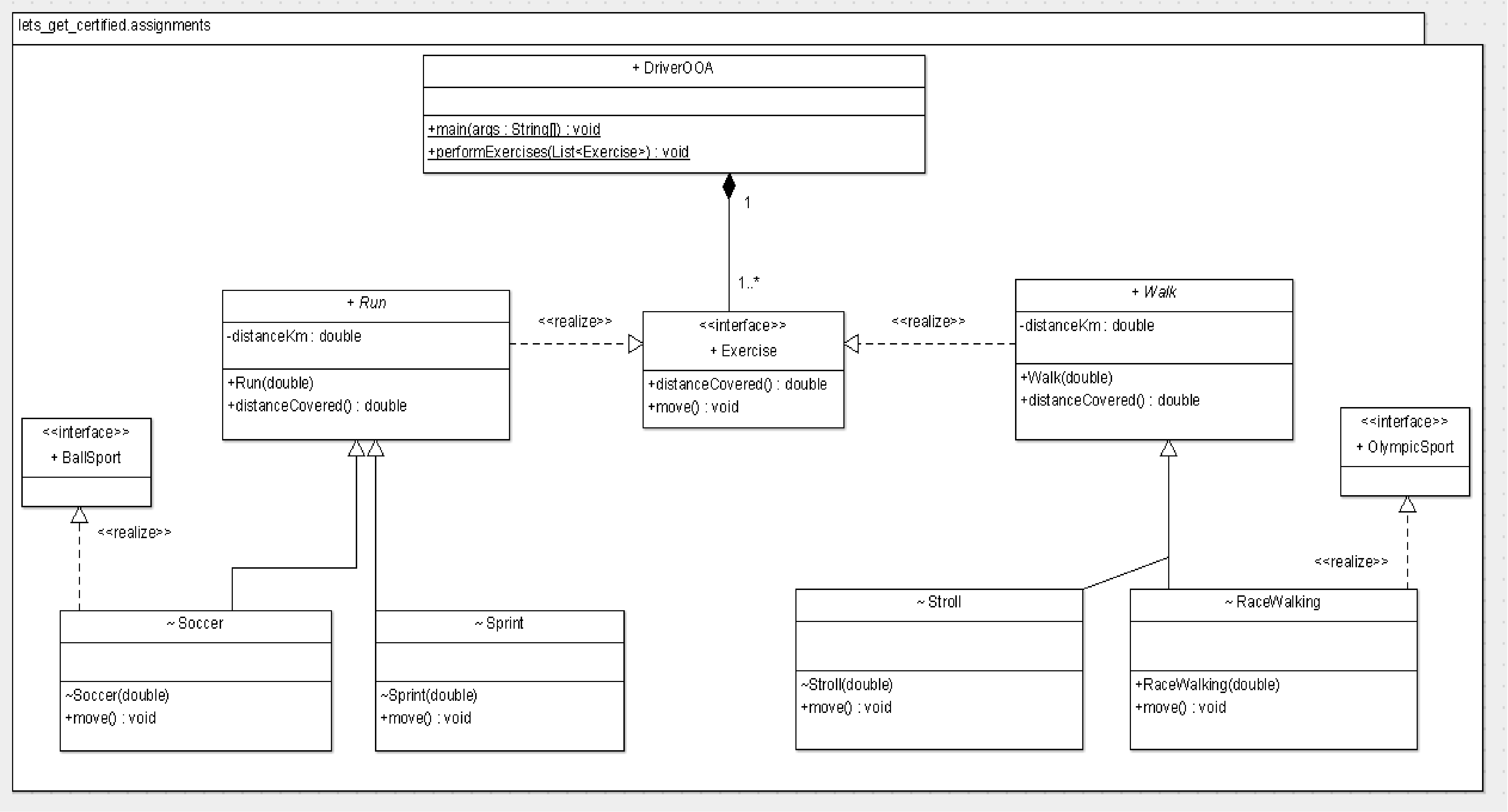
OOA exercise



1. Code the UML.
    a. *Run* and *Walk* are both in italics in the UML i.e. they are abstract.
        i. constructors initialise the instance variable
        ii. *distanceCovered()* returns the *distanceKm* instance variable
    b. *Stroll* and *RaceWalking*
        i. constructors pass the parameter up to the parent constructor
        ii. *move()* method just outputs “moving” preceded by the class name, for example: *Stroll::moving()*
    c. *Soccer* and *Sprint* are as per *Stroll* and *RaceWalking*

2. *DriverOOA* class:
    a. *main()* method
        i. create a *List* implemented by *ArrayList*; use type inference/diamond operator.
        ii. create a *Soccer* object where the *distanceKm* is 12.5 (km); refer to the object using an *Exercise* reference; add to the list.
        iii. create a *Sprint* object where the *distanceKm* is .1 (km); refer to the object using an *Exercise* reference; add to the list.
        iv. create a *Stroll* object where the *distanceKm* is 10 (km); refer to the object using an *Exercise* reference; add to the list.
        v. create a *RaceWalking* object where the *distanceKm* is 5 (km); refer to the object using an *Exercise* reference; add to the list.
        vi. invoke the method *performExercises()* passing down the list
    b. *performExercises()* method
        i. using an enhanced-for loop, loop through each exercise
            1. if the exercise is a *BallSport*, output the (simple) class name i.e. *refName.getClass().getSimpleName()*.
            2. if the exercise is an *OlympicSport*, output the (simple) class name