**The following Classes are ones that are already given:**

**Interface:**

BadCyclingPortalimpl.java

* BadCyclingPortalImpl
  + Calls a lot of methods that interact with the interface

BadMiniCyclingPortalimpl.java

* BadMiniCyclingPortalImpl

CheckpointType.java

* Enum CheckpointType

CyclingPortal.java

* Interface CyclingPortal

DuplicatedResultException.java

* DuplicatedResultsException

IDNotRecognisedException.java

* IDNotRecognisedException

IllegalNameException.java

* IllegalNameException

InvalidLengthException.java

* InvalidLengthException

InvalidLocationException.java

* InvalidLocationException

InvalidNameException.java

* InvalidNameException

InvalidStageTypeException.java

* InvalidStageTypeException

InvalidStageStateException.java

* InvalidStageStateException

MiniCyclingPortal.java

* MiniCyclingPortal (Interface)

**The following classes are classes that we have come up with ourselves:**

Interface Rider/Team:

* Provides a blueprint for classes representing riders and teams in cycling races.
* Includes methods and properties common to both such as name, nationality, team, etc

Class Riders:

- String name

- Nationality

- Team

Class Teams:

- Team name

- Number of riders

- Constructor, Getters and setters

Class Races:

-Race name

- Racers inherit from teams

- getters and setters for properties

- Methods to manage stage and teams

Class Stage:

* Stage number/type
* Stage distance
* List of checkpoints
* Constructor, getters and setters

Class Competitions:

* Competition Type
* Ranking
* Constructor, getters, setters

Interface results:

* Result type
* More stuff on results

Class Stage:

* Stage Type
* Distance
* Checkpoints (Checkpoint type is already an enum value)

Abstract Competition Class:

* To represent a competition within a race (General Classification, Points Classification, or Mountain Classification) – these will be subclasses of competition
* Provides common properties and methods for different types of competitions
* Subclasses can specialize for each specific competition type

RaceResult interface:

* Allows flexibility of implementing different types of race results, providing a structure for storing and accessing race outcome data.

If there is anything I should add, lmk.