Design document

s3582779 Yipeng Zhang
s3528894 Yanjie Zhang

First page is explanation of the design, second page is our class diagram and contribution distribution, experience and harvest.
Github screenshot is in the zip.

1. **Explain how your design will be able to store the information of games, athletes and user predictions.**

   We create two texts called “Athlete list” and “Official list” in the Data folder so that we can store default athletes and official data.

   We use construction to give game, athletes and officials detailed information and store each information in the arraylist including specific detailed data such as game id, athletes points and final points of every athletes. New an object of the athlete class and use mutator to get name, age, state and type of the athlete.

   Prediction works as that input a number meaning as an athlete, and make it as a parameter and compare it with the real winner. In a method called showPredict, judge the predicted player is 1st or not.

2. **Explain how your class hierarchy will forbid a user from creating a “generic” type of participant (i.e. not a athlete nor an official)**

   Athlete and official are two different class but both inheriting the user class. While user class is an abstract class, nothing can be created by the user class. Athlete and official class have different construction, so, if an user want to create a new role, it must be an athlete or an official, but never be a “generic” type.

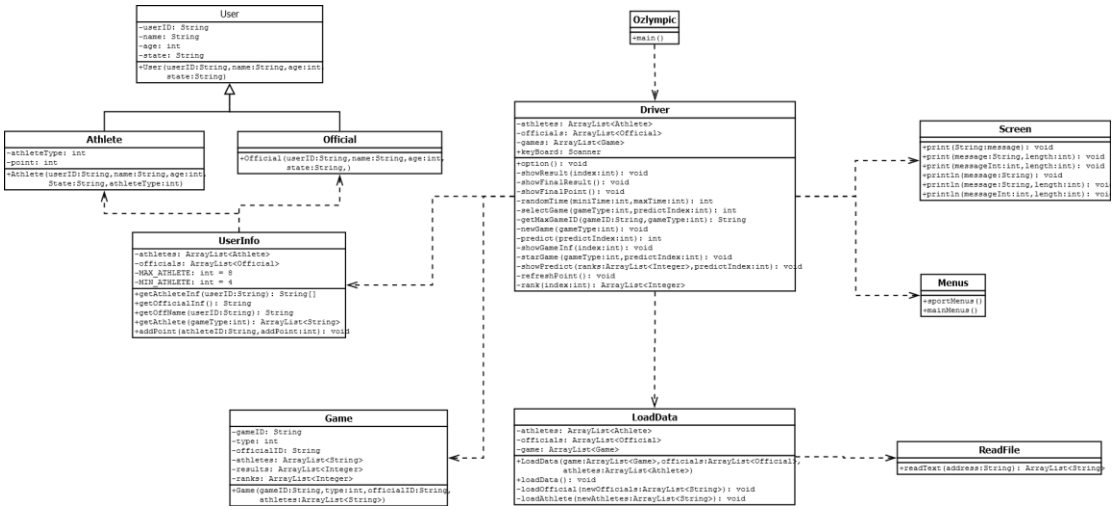
3. **Explain the process by which your program will maintain a game and give correct score to athletes according to their performance.**

   Firstly, we use selectGame method to chose the specific game, and then create a new game(getAthlete method will chose 8 random athlete who meet the requirement of choosing athlete for a game) . After starting the game(call starGame method) it will automatically distribute results among every athletes(setResult method) and use rank method to give every athletes a rank in this game. Finally, we use refreshpoint method to add correct points for first three place athletes.

4. **Explain how a user prediction can be checked with the actual game results.**
   We make the prediction number(represent an athlete) as a parameter and compare it

with the real winner. In a method called showPredict, judge the predicted player is 1st or not. If the predicted athlete get the 1st, system will output a congratulation message on the screen.

Class diagram.



Contribution and experience:

Yipeng Zhang: User class, Athlete class, Official class, Game class loadData class, readFile class, UserInfo class, Screen class and Program construction design.

Even this assignment is a small program, we also used a lot of object-oriented concept. The User class is designed to an abstract class, which is inherited by Athlete class and Official class. We try to logically split the whole program to different class. For example, Screen is made as a User Interface. ReadFIle and LoadData can read data from text file and organize them as a String, which is Input Interface. UserInfo class contain all methods about user information. No matter read or write data about athletes and officers, Driver class can use it. These designs can make a good balance with coupling and cohesion. Meanwhile, it gives me insight into how to design an object-oriented project.

Moreover, I have learned how to cooperate with other partners through GitHub and email. There were some misunderstand about program constructions and methods, but we fix all of them. Also, I have learned how to avoid these problems as much as possible. I believe it a crucial knowledge and skill for team work.

Yanjie Zhang: Driver class. Program construction design.

Through this assignment, I happily cooperated with Yipeng Zhang. We divided the task into two parts. I'm responsible for the driver class which is important to implement the major function in this program. I've got lots of experience on coding and program construction. About coding skills, I exercised to implement the

requirement and whole logic in the program. About program construction, I've tried to make the program like 'low coupling and high cohesion'. Because I always need to call some methods from other class, I need to understand the whole construction and make sure that my partner also present an obvious coding. That is really a challenge in stead of coding and programming.

About teamwork, I think it was a very fantastic experience because we made a perfect cooperation with almost no problem. By using github and email, we kept communication and progress on the coding part. It is really a useful experience for me to study Java. Communication is so important that I don't need to discover the trouble by myself. We can handle lots of problem together and find it is easier to fix them.