

Jogging Club Recommendation System

In this assignment, you will have the opportunity to practice writing a Java program to act as a recommendation system for a local jogging club. The goal of your program is to match pairs of runners from the jogging club who would make good running partners, based on a predefined set of criteria (outlined below). You will begin by writing a Java class, called "ClubMember" to represent and describe a member of the jogging club.

A ClubMember object will have the following attributes:

the member's name (a string)

the member's age (an integer)

preferred distance to run in kilometers (an integer – 5, 10, 15, 30)

intensity (boolean – this value is false if the member prefers to do a "walk/jog" where they run for a distance and then take a break by walking. The value will be true if the member prefers to run the entire time).

time of day when the member prefers to run (this will be a string: "morning", "afternoon" or "evening")

The constructor for your ClubMember class will take five parameters. The parameters will correspond to the instance variables described above. Please ensure that your constructor accepts the parameters in the same order that they are listed above: name, age, distance, intensity and time of day.

You will make decisions on who should be paired to run together based on the following criteria:

- Members would be matched together if they prefer to run at the same time of day (morning, afternoon or evening).
- Members should be within the same age range. For this assignment, the age ranges will be as follows: 20-24, 25-29, 30-34, 35-39, 40-44, 45-49, 50-54, 55-59, 60-64, 65-69, 70+.

- Members should only be grouped together if they prefer to run the same distance.
- A member who prefers to do “walk/jogs” should not be grouped with a member who would rather run the entire distance without pausing for a walk.

You should include a toString method in your class – this method will return a string describing a club member in a format like the following examples:

Bob Jones: 22, 10, true, afternoon

Betty Jones: 79, 5, false, morning

REQUIRED METHODS

1. Your class should also contain an instance method called matchRunners. If BobJones and BettyJones are clubMember objects, BobJones.matchRunners(BettyJones) would return true if the two runners would make good running partners, otherwise false. The matching should go both ways, that is, BettyJones.matchRunners(BobJones) should return the same value.
2. Your class should contain a method called readRunners that will read in the members' information from the provided [text file](#).
3. Sometimes runners change their distance preferences. Be sure to include a method called setDistance that accepts a runner's name and their new preferred distance as its parameters. For example: BobJones.setDistance(15) would set Bob Jones's new preferred running distance to 15K.
4. Write a method called FindMatches. This method will accept a ClubMember object and will return a list of all possible running partner matches for this ClubMember based on the matching criteria outlined above.

5. You will also write a void, static instance method called `displayPairs`. This method will accept an array of `clubMember` objects as its parameter and will print a listing of the pairs of runners that would make good running partners. Please be sure that each pair is printed at most one time and that no individuals are paired with themselves. You can decide on how the output will appear. Be sure that the output is easy to understand and organized

ERROR CHECKING & INFORMATION HIDING

Error Checking

You are required to check for the following errors in user input.

- If an age less than 20 is passed into the constructor, throw an exception and print out the message ("Illegal age – runners can only be a member of the club if they are 20+ years old").
- If a time of day other than "morning", "afternoon" or "evening" is passed into the constructor, throw an exception and print out the message ("Illegal time of day provided").
- If a distance other than 5, 10, 15 or 30 is provided to the constructor, once again throw an exception. The exception message should be ("Illegal distance provided").

Information Hiding

Marks will be allocated according to how well you have implemented the principles of information hiding in the development of your class.

Reminders:

- All instance and class variables should be private. This ensures that other classes are not able to modify the values of these attributes.

- Please ensure that your class includes accessor ("getter") and mutator ("setter") methods for each of the instance variables.
- Constants should be private unless they need to be used by other classes.
- Any method that another class should be able to call, and access should be public. In this assignment, another class will need to call the required methods, therefore, these methods should be made public.
- Any "helper methods" should be private. A "helper method" is a method that you write to help implement the required methods.
- You will need to write an exception class called "IllegalClubMember". This exception class only needs the one constructor that accepts a String type message.
- Throw an "IllegalClubMember" exception object if an attempt is made to set or alter any of the attributes to illegal values.
- Please ensure that your classes include accessor ("getter") and mutator ("setter") methods for each of the instance variables.