

Quiz 2

Problem

Write an object oriented program that allows users to manage the members of Fitness Center. The sport center has more than one person and each person may register either as a personal trainer or as a member.

The program should store all of the information (i.e. id (**must start at 1**), name, surname, height, weight, etc.) as shown in Figure 1.

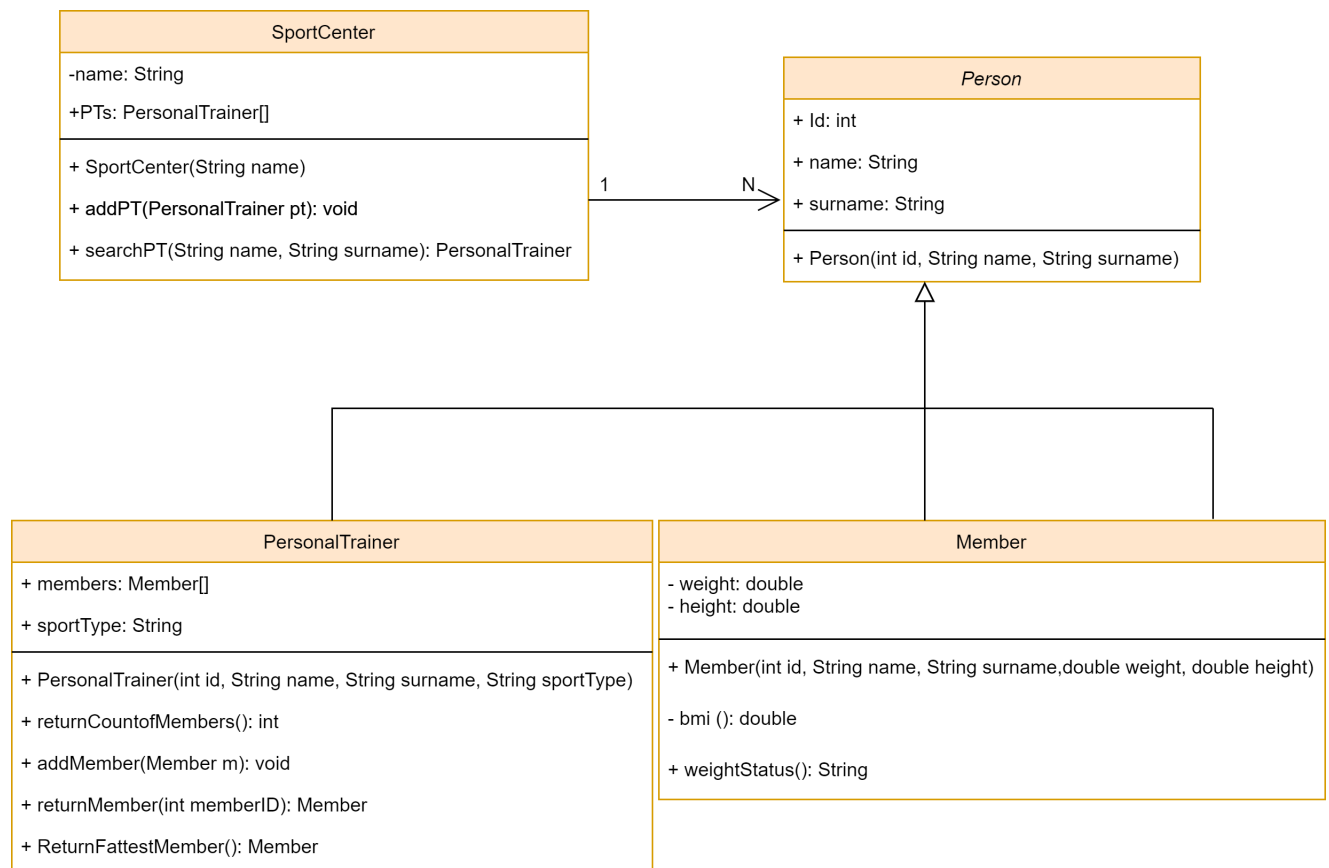


Figure 1: UML class diagram of Sport Center

Create a **Person** class with:

- Several attributes: id, name, surname.
- One constructor method which receives all attributes as parameters.

Create a **PersonalTrainer** class with:

- Several attributes: Member[] members, sportType.
- One constructor method which receives all attributes as parameters (id, name, surname, sportType).
- addMember(Member m): adds members to the specified PT.
- returnCountofMembers(): returns the count of specified PT's members.
- returnMember(int memberID): returns Member instance.
- ReturnFattestMember(): Returns the fattest member PT has.

Create a **Member** class with:

- Several attributes: heigth, weight. These attributes must be private.
- One constructor method which receives all parameters (id, name, surname, weight, height).
- Get / Set methods
- bmi() method: calculates Body Mass Index of the member. The formula for BMI index is shown in Figure 2. W represents weight, and H represents height of members.

$$BMI = \frac{W}{H^2}$$

Figure 2: The formula of BMI index

- weightStatus() method: returns the weight status of the member as String by calling BMI method.

| Body Mass Index | Weight Status |
|-----------------|---------------|
| Smaller 18.5 | Thin |
| 18.5 – 24.9 | Normal |
| 25 – 29.9 | Fat |
| 30 and up | Obese |



Create a **SportCenter** class with:

- Several attributes: name of the sport center, PersonalTrainer[] PTs.
- One constructor which receives name as parameter
- addPT(PersonalTrainer pt) method: adds personal trainer to the sport center
- searchPT(String name, String surname) method: returns an instance of type "PersonalTrainer".

1 Grading Policy

| Task | Point |
|--------------------------------------|-------|
| ReturnFattestMember() method | 20 |
| weightStatus() method | 10 |
| searchPT() method | 20 |
| addPT() method | 20 |
| returnMember() method | 10 |
| returnCountofMembers() method | 20 |
| Total | 100 |

2 Important Notes

- We will give you Test.java file. In this file each test is given. You should create your class and functions according to method calls. Your function's parameters should be appropriate for function calls.
- You shouldn't change the name or return type of functions. You have to use them as specified in the pdf. However, you can add auxiliary functions and attributes if you want.
- Please test your code before submitting it on dev server. The submissions which can not compile will not be evaluated.
- In order to test your program, you should follow the following steps:
 - Upload your java files to your server account (dev.cs.hacettepe.edu.tr)
 - Compile your code (javac *.java)
 - Run your program (java Test)
 - Control your output data (output.txt) as shown in Figure 3.

```
You passed addPT() method test
You passed searchPT() method test
You passed weightStatus() method test
You passed ReturnFattestMember() method test
You passed returnMember() method test
You passed returnCountofMembers() method test
```

Figure 3: Output of program

- Save all work until the quiz is graded.
- Do not miss the deadline. Submission will be end at 02/11/2023 23:00, the system will be open until 23:00:00. The problem about submission after 23:00 will not be considered.

