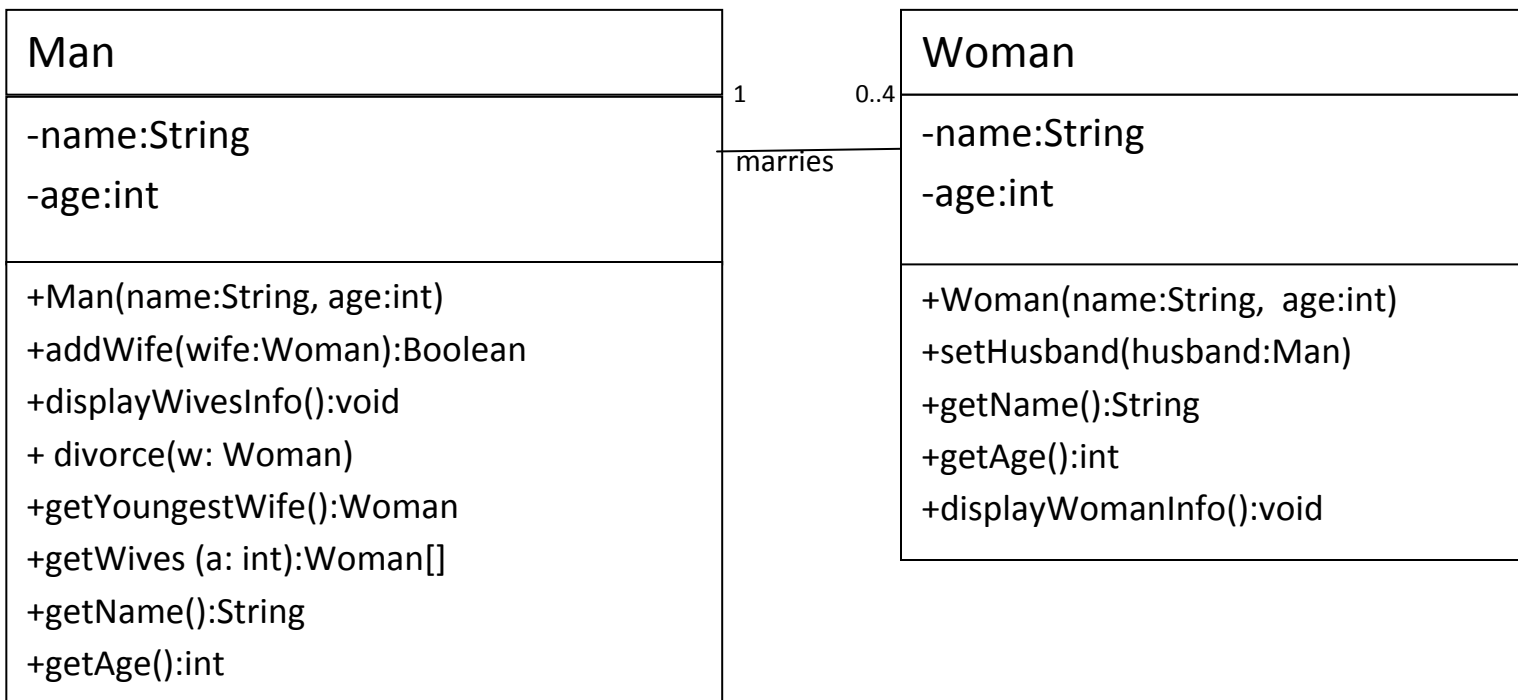


Create the classes along with the functionality given in the following UML Diagram. To understand the problem, please refer to the description given after the diagram.



**Woman Class:**

- Attributes:
  - **name**: the name of the Woman.
  - **age**: age of the Woman
- Methods:
  - **Woman(name: string, age: int)**: constructor. Assigns given values to the attributes and assigns *null* to *husband*.
  - **setHusband(husband:Man)**: sets the husband of the woman.
  - **displayWomanInfo()**: this method displays all the attributes of the Woman along with the name and age of Husband if woman is married.
  - **getters**: return the values of the corresponding attributes

### **Man Class:**

#### ○ Attributes:

- **name**: the name of the Man.
- **age**: age of the Man

#### ○ Methods:

- **Man(name:String, age:int)** : constructor
- **addWife(wife:Woman)**: this method associates a new wife to the man. It returns true if the **wife** is associated false otherwise.
- **displayWivesInfo()**: displays the detail(name and age) of every wife which is married to the man and in case if man is unmarried , it will display “Ohh..He is unmarried”.
- **divorce(w: Woman)**: this method removes the Woman w from the wives of the man.
- **getYoungestWife()**: returns the youngest wife of the man.
- **getWives(a: int)**: this method returns an array containing all the wives of the man whose age is less than **a**.
- **getters**: return the values of the corresponding attributes

### **HINT: How to Solve Association**

**Man** class will have an attribute array **arrwives** of type **Woman** along with a **counter** and **Woman** class will an attribute **husband** of type **Man**.