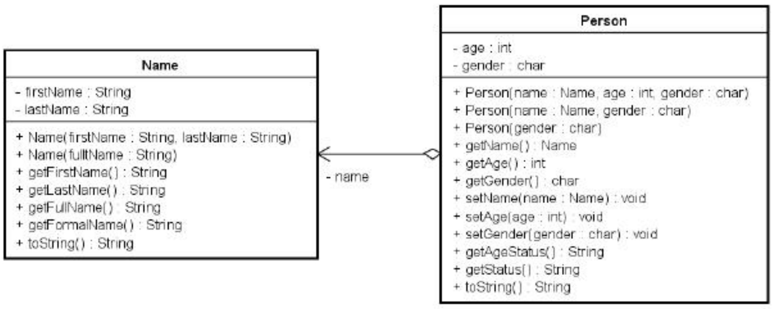
Exercise: Person, version 4

Create a new module and name it Person_v4 and copy classes Person, Name and MyDate.

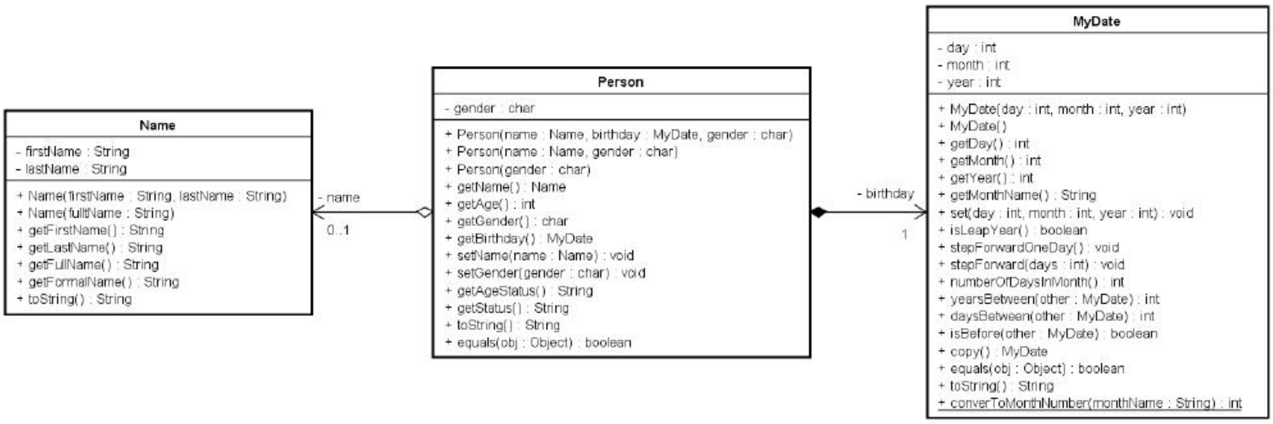
Part 1: Person and Name (Aggregation) – [Video solution]


Name
- firstName : String
- lastName : String
+ Name(firstName : String, lastName : String)
+ Name(fullName : String)
+ getFirstName() : String
+ getLastName() : String
+ getFullName() : String
+ getFormalName() : String
+ toString() : String
- name
Person
- age : int
- gender : char
+ Person(name : Name, age : int, gender : char)
+ Person(name : Name, gender : char)
+ Person(gender : char)
+ getName() : Name
+ getAge() : int
+ getGender() : char
+ setName(name : Name) : void
+ setAge(age : int) : void
+ setGender(gender : char) : void
+ getAgeStatus() : String
+ getStatus() : String
+ toString() : String

Find the version of class Person with an aggregation to class Name or modify your Person class such that the person has a name of type Name (and not of type String). Modify the constructors, the getName and the setName methods such that you use type Name instead of String.

If class Name did not have a toString method, then implement it. Why do you think this is a good idea?

Part 2: Person and MyDate (Composition) – [Video solution]


Name
- firstName : String
- lastName : String
+ Name(firstName : String, lastName : String)
+ Name(fullName : String)
+ getFirstName() : String
+ getLastName() : String
+ getFullName() : String
+ getFormalName() : String
+ toString() : String
- name
0..1
Person
- gender : char
+ Person(name : Name, birthday : MyDate, gender : char)
+ Person(name : Name, gender : char)
+ Person(gender : char)
+ getName() : Name
+ getAge() : int
+ getGender() : char
+ getBirthday() : MyDate
+ setName(name : Name) : void
+ setGender(gender : char) : void
+ getAgeStatus() : String
+ getStatus() : String
+ toString() : String
+ equals(obj : Object) : boolean
- birthday
1
MyDate
- day : int
- month : int
- year : int
+ MyDate(day : int, month : int, year : int)
+ MyDate()
+ getDay() : int
+ getMonth() : int
+ getYear() : int
+ getMonthName() : String
+ set(day : int, month : int, year : int) : void
+ isLeapYear() : boolean
+ stepForwardOneDay() : void
+ stepForward(days : int) : void
+ numberOfDaysInMonth() : int
+ yearsBetween(other : MyDate) : int
+ daysBetween(other : MyDate) : int
+ isBefore(other : MyDate) : boolean
+ copy() : MyDate
+ equals(obj : Object) : boolean
+ toString() : String
+ converToMonthNumber(monthName : String) : int

Modify class Person:

a) Delete instance variable age and insert another instance variable birthday (type MyDate)

b) Modify the three-argument constructor to take birthday instead of age as a parameter. Remember to store a copy (composition)

c) Modify the two-argument and one-argument constructor to create a date representing current date (using the zero-argument constructor in MyDate)

d) Add a method getBirthday returning a copy of the birthday instance variable.

e) Modify method getAge to calculate the age using method yearsBetween in class MyDate.

f) Do not add a method to set the birthday (why not?)

g) Modify methods getAgeStatus and getStatus to use method getAge now you do not have an age instance variable any longer.

h) Modify toString to include the birthday in the string being returned.

i) If you implemented an equals method then update this to include birthday in the check.