

## CSCI 1302

### # Homework A2: Sports N' Stats System

Demonstrated proficiency: Inheritance, polymorphism, explicit casting, regex, toString methods, overriding methods, sorting, use of "super" for superclass' constructors and methods, LocalDate, ChronoUnit and Period classes.

Implement the Athlete, Swimmer, Runner, and AthleteRoster classes below. Each class must be in separate file. Draw an UML diagram with the inheritance relationship of the classes.

#### 1. The Athlete class

- a. All class variables of the Athlete class must be private.
- b. is a class with a single constructor, **Athlete(String lName, String fName, int birthYear, int birthMonth, int birthDay, char gender)**. All arguments of the constructor should be stored as class variables. There should only be getter methods for first and last name variables. **There should be no getters or setters for birthYear, birthMonth, birthDay.**
- c. Getter and setter methods for the gender of the athlete. The method **setGender** accepts a character and store it as a class variable. Characters 'm', 'M' denotes male, 'f' or 'F' denotes male or female. Any other char will be treated as undeclared. The method **getGender** return either the word "male" or "female" or "undeclared".
- d. The method **computeAge()** takes no arguments and returns the athletes computed age (as of today's date) as a string of the form "X years, Y months and Z days". **Hint: Use the LocalDate and Period classes.**
  - i. e.g. "21 years, 2 months and 3 days".
  - ii. e.g. "21 years and 3 days".
  - iii. e.g. "21 years and 2 months".
  - iv. e.g. "21 years".
  - v. e.g. "2 months and 3 days".
- e. The method public long daysSinceBirth() takes no arguments and returns a long which is the number of days since the athlete's date of birth to the current day. **Hint: Use the LocalDate and ChronoUnit classes.**
- f. The **toString** method returns a string comprised of the results of **getFname**, **getLName** and **computeAge**. E.g.
  - i. "Bunny Bugs is 19 years and 1 day old"
- g. The **equals** method returns true if the name and date of birth of this athlete and the compared other athlete are the same, otherwise return false.

## 2. The Swimmer class

- a. All class variables of the Swimmer class must be private.
- b. inherits from the Athlete class and has a single constructor, **Swimmer(String lName, String fName, int birthYear, int birthMonth, int birthDay, char gender, String team)**. There should be a class variable for team. There should be a getter method for team.
- c. The Swimmer class stores swim events for the swimmer. There should be a class variable for events. A swimmer participates in one or more of these events. The **addEvent** method is **overloaded** and either adds a single event for the swimmer **public boolean addEvent(String singleEvent)** or adds a group of events for the swimmer **public boolean addEvent(ArrayList<String> multiEvents)**. Each event is of type String. Duplicate events are not stored and return false if duplicate found.
- d. There should be a getter method that returns the class variable events.
- e. The **overridden toString** method returns a string comprised of the concatenation of the parent's **toString** return plus " and is a swimmer for team: XXXXX in the following events: YYYY; ZZZZZZ." E.g.
  - i. "Missy Franklin is 24 years and 4 months old and is a swimmer for team: Colorado Stars. She participates in the following events: [100m freestyle, 100m backstroke, 50m backstroke]"

## 3. The Runner class

- a. All class variables of the Runner class must be private.
- b. inherits from the Athlete class and has a single constructor, **Runner(String lName, String fName, int birthYear, int birthMonth, int birthDay, char gender, String country)**. There should be a class variable for country. There should be a getter method for country.
- c. The Runner class stores race events for the runner. There should be a class variable for events. Each event is a Distance. The list of valid events is given below:  
**M100, M200, M400, M3000, M5000, M10000**. A runner participates in **one or more** of these events. The **addEvent** method is **overloaded** and either adds a single event for the runner **public boolean addEvent(String singleEvent)** or adds a group of events for the runner **public boolean addEvent(ArrayList<String> multiEvents)**. Each event is of type String. Duplicate events are not stored and return false if duplicate found.

- d. There should be a getter method that returns the class variable events.
- e. The **toString** method returns a String in the form: " AAA BBBB is XX years, YY months and ZZ days. He is a citizen of CCCCCC and is a DDDDDDD who participates in these events: [MJ00, MK00, ML00]". If she does not participate in M3000 or M5000 or M10000 then she is a **sprinter**. If she does not participate in M100 or M200 or M400 then she is a **long-distance runner**. Otherwise she is a **super athlete**. E.g.
  - i. "Bunny Bugs is 19 years and 1 day old. His is a citizen of USA and is a long-distance runner who participates in these events: [M10000]"

#### 4. The AthleteRoster class

- a. All class variables of the AthleteRoster class must be private.
- b. Does not inherits from the Athlete class. The AthleteRoster class has a single constructor, **AthleteRoster(String semester, int year)**. There should be class variables for semester and year. There should be getter methods for semester and year.
- c. The AthleteRoster class has only one method for adding Athlete to the roster, by using the **boolean addAthlete(Athlete a)** method. The method returns true if the athlete was added successfully, it returns false if the athlete object already exists in the roster and therefore was not added.
- d. Your AthleteRoster class will have only one class level data structure, an ArrayList, for storing athlete objects.
- e. The **String allAthletesOrderedByLastName()** method returns a string object with of all the names of the athletes (Swimmers, Runners, etc.) in ascending order of last names(a-z).
- f. The **String allAthletesOrderedByAge()** method returns a string object with of all the names of the athletes (Swimmers, Runners, etc.) in descending order of age(100-0).
- g. The **String allRunnersOrderedByNumberOfEvents()** method returns a string object with of all the names of the Runners only in ascending order of number of events they participate in (0-100).

Complete the code before the due date. **Submission of the completed eclipse project is via github link posted on the class page.** Add your UML drawing to the github repo.

---

Example output:

\_\_\_\_\_Example from AthleteDriver shown below

```
Gender is undeclared
Gender is female
Gender is male
ComputeAge method says 19 years and 4 days
First name is : Duck
DaysSinceBirth method says 6943 days
Last name is : Daffy
Output of our toString correct?:
Duck Daffy is 19 years and 4 days old
=====

Did we add M10000 successfully?: true
Did we unsuccessfully try to add M10000 again?: false
Did we successfully add multiple events?: true
Did we unsuccessfully try to add multiple events?: false
How many events does Bugs participate in?: 3
Gender is male
Output of our toString correct?:
Bunny Bugs is 19 years and 3 days old. His is a citizen of USA and is a super
athlete who participates in these events: [M10000, M100, M3000]
=====
```

\_\_\_\_\_Example from AthleteDriver shown below

```
Did we add 100m backstroke successfully?: true
Did we unsuccessfully try to add 100m backstroke again?: false
Did we successfully add multiple events?: true
Did we unsuccessfully try to add multiple events?: false
How many events does s1 participate in?: 3
Gender is female
Output of our toString correct?:
Missy Franklin is 24 years, 4 months and 1 day old and is a swimmer for team:
Colorado Stars. She participates in the following events: [200m backstroke, 200m
freestyle, 100M BACKSTROKE]
```

=====

List of Runners Ordered by Number of Events They participate In

=====

```
Usain Bolt 2
Bunny Bugs 3
Florence Griffith 6
```

=====

List of Athletes Ordered by their Age

=====

```
Griffith, Florence 59 years, 8 months and 21 days
Bolt, Usain 33 years and 21 days
Franklin, Missy 24 years, 4 months and 1 day
Ruele, Naomi 22 years, 7 months and 29 days
Daffy, Duck 19 years and 4 days
Bugs, Bunny 19 years and 3 days
```

=====

\_\_\_\_\_Example from AthleteDriver shown below

List of Athletes Ordered by their Last Name

=====

Bolt, Usain

Bugs, Bunny

Daffy, Duck

Franklin, Missy

Griffith, Florence

Ruele, Naomi

=====