CSE214

HOMEWORK 1- Spring 2023

HOMEWORK 1 - due Tuesday, February 7th no later than 7:00PM

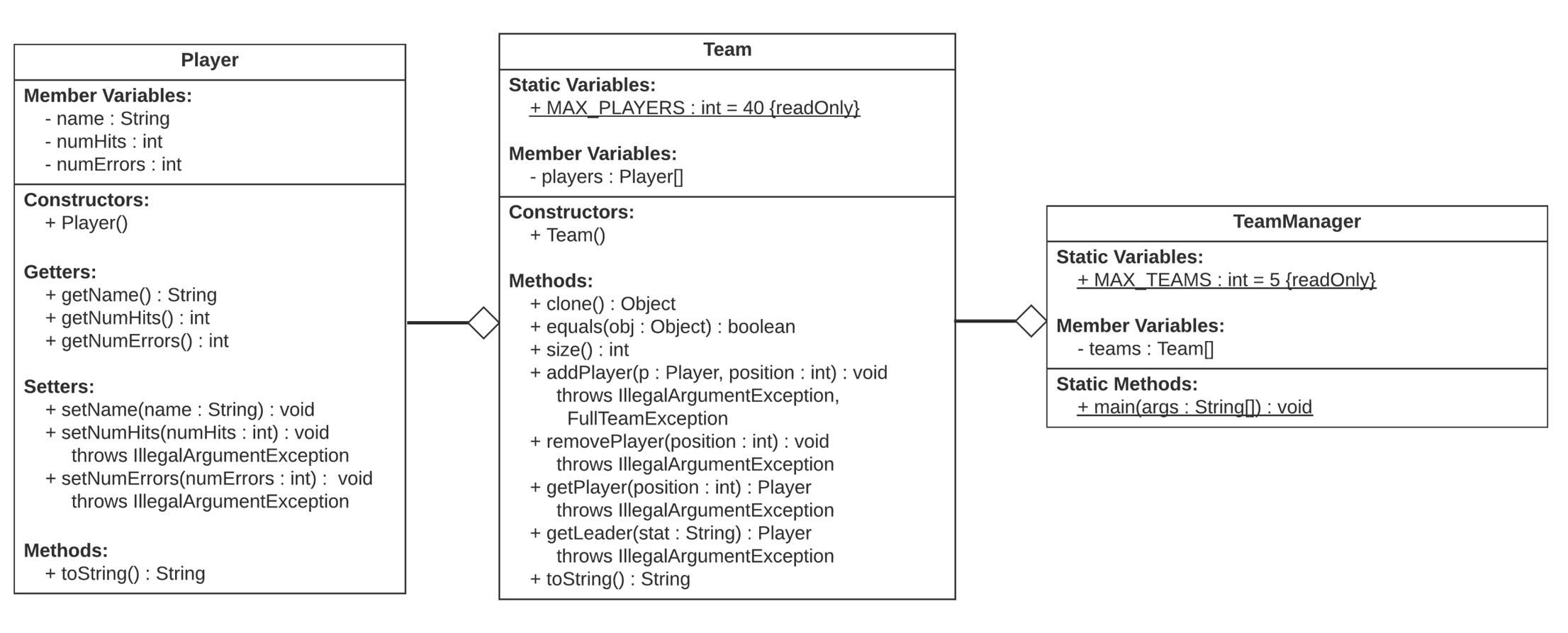
REMINDERS:

- **Use of a package is optional. If you wish to use it, make sure to name it "hw1" (all in lower case). Otherwise, you will lose points.**
- **Be sure your code follows the coding style for CSE214.**
- **Make sure you read the warnings about academic dishonesty. *Remember, all work you submit for homework or exams MUST be your own work.***
- **Login to your grading account and click "Submit Assignment" to upload and submit your assignment.**
- **You are not allowed to use ArrayList, Vector or any other Java API Data Structure classes to implement this assignment.**
- **You may use Scanner, InputStreamReader, or any other class that you wish for keyboard input.**

In this assignment, you will be required to write a Java program to keep track of a baseball team's statistics. A team consists of up to 40 players, each of whom has a certain number of hits and errors.

UML

The UML Diagram for all the classes specified below is as follows:



Required Classes

The following sections describe classes which are required for this assignment. Each section provides a description and the specifications necessary to complete each class. If you feel that additional methods would be useful, feel free to add them during your implementation as you see fit. However, all the variables and methods in the following specifications must be included in your project.

1. Player

Write a fully-documented class named Player which contains the player's name (String), number of hits (int), and number of errors (int). You should provide accessor and mutator methods for each variable as well as a default constructor. For the mutator of the hits and errors variables, you should throw an exception if the new value is less than 0. You should also provide a toString() method that returns a printable representation of the player and his statistics (hits and errors).

2. Team

Write a full-documented class named Team that stores all Player objects that belong to a particular team. The Player objects should be stored in an array. There should be a maximum of 40 Player objects allowed, a number which should be defined as a final variable. The class will be based on the following ADT specification:

- **public static final int MAX_PLAYERS = 40;**

- **public Team()**
  - ***Brief:***
    - **Construct an instance of the Team class with no Player objects in it.**
  - ***Postconditions:***
    - **This Team has been initialized to an empty list of Players.**

- **public Object clone()**
  - ***Brief:***
    - **Generate a clone of this Team.**
  - ***Returns:***
    - **The return value is a clone of this Team. Subsequent changes to the clone will not affect the original, nor vice versa. Note that the return value must be typecast to a Team before it can be used..**

- **public boolean equals(Object obj)**
  - ***Brief:***
    - **Compare this Team to another object for equality.**
  - ***Parameters:***
    - **obj - an object to which this Team is compared**
  - ***Returns:***
    - **A return value of true indicates that obj refers to a Team object with the same Players in the same order as this Team. Otherwise, the return value is false. If obj is null or it is not a Team object, then the return value is false.**
  - ***Note:***
    - **When comparing equality between two Player objects, you must verify that their names, hits, and errors are all the same. Using the == operator will simply check to see if the two variables refer to the same Player object, which does not take into consideration that two different Player objects can actually represent the same person. To solve this problem, you can either check that each of the properties of the two objects are the same (name, hits, errors) inside of this method, or you may simplify this process by implementing an equals method (similar to this one) for the Player class.**

- **public int size()**
  - ***Brief:***
    - **Determines the number of Players currently in this Team.**
  - ***Preconditions:***
    - **This Team object has been instantiated.**
  - ***Returns:***
    - **The number of Players in this Team.**

- **public void addPlayer(Player p, int position)**
  - ***Brief:***
    - **Adds a Player to the team at the indicated position in the lineup.**
  - ***Parameters:***
    - **p - the new Player object to add to this Team.**
    - **position - the position in the lineup where the Player will be inserted .**
  - ***Preconditions:***
    - **This Team object has been instantiated**
    - **1 ≤ position ≤ players_currently_in_team + 1.**
    - **The number of Player objects in this Team is less than MAX_PLAYERS.**
  - ***Postconditions:***
    - **The new Player is now stored at the desired position in the Team. All Players that were originally in positions greater than or equal to position are moved back one position.**
    - **E.g. If there are 5 Players in a Team, positions 1-5, and you insert a new Player at position 4, the new Player will now be at position 4, the Player that was at position 4 will be moved to position 5, and the Player that was at position 5 will be moved to position 6).**
  - ***Throws:***
    - **IllegalArgumentException - Indicates that position is not within the valid range..**
    - **FullTeamException - Indicates that there is no more room inside of the Team to store the new Player object.**
  - ***Note:***
    - **Position refers to the position in the Team lineup and not the position inside the array.**
    - **Inserting a Player to position (players_currently_in_team + 1) is effectively the same as adding a Player to the end of the Team.**

- **public void removePlayer(int position)**
  - ***Brief:***
    - **Removes a Player from the team at the indicated position in the lineup.**
  - ***Parameters:***
    - **position - the position in the lineup from which the Player is to be removed.**
  - ***Preconditions:***
    - **This Team object has been instantiated**
    - **1 ≤ position ≤ players_currently_in_team .**
  - ***Postconditions:***
    - **The Player at the desired position in the Team has been removed.**
    - **All Players that were originally in positions greater than or equal to position are moved forward one position.**
    - **E.g. If there are 5 Players in a Team, positions 1-5, and you remove the Player at position 4, the Player that was at position 5 will be moved to position 4.**
  - ***Throws:***
    - **IllegalArgumentException - Indicates that position is not within the valid range..**
  - ***Note:***
    - **Position refers to the position in the Team lineup and not the position inside the array.**

- **public Player getPlayer(int position)**
  - ***Brief:***
    - **Returns a reference to a player in the lineup at the indicated position.**
  - ***Parameters:***
    - **position - the position in the lineup from which the Player is to be retrieved.**
  - ***Preconditions:***
    - **This Team object has been instantiated.**
  - ***Returns:***
    - **The Player from the given index.**
  - ***Throws:***
    - **IllegalArgumentException - Indicates that position is not within the valid range.**
  - ***Note:***
    - **Position refers to the position in the Team lineup and not the position inside the array.**

- **public Player getLeader(String stat)**
  - **Note: If you implement this method using 'int' stat (0 or 1 indicating hits or errors), this is acceptable as well.**
  - ***Brief:***
    - **Return the Player with the best value in the given statistic ("hits" or "errors").**
  - ***Parameters:***
    - **stat - either "hits" or "errors".**
  - ***Preconditions:***
    - **This Team object has been instantiated.**
  - ***Returns:***
    - **The Player with the best stat.**
  - ***Throws:***
    - **IllegalArgumentException - Indicates that indicated stat was neither "hits" nor "errors".**
  - ***Note:***
    - **Remember that HIGHER hits are good, whereas LOWER errors are good.**

- **public void printAllPlayers()**
  - ***Brief:***
    - **Prints a neatly formatted table of each Player in the Team on its own line with its position number as shown in the sample output.**
  - ***Preconditions:***
    - **This Team object has been instantiated.**
  - ***Preconditions:***
    - **A neatly formatted table of each Player in the Team on its own line with its position number has been displayed to the user.**
  - ***Hint:***
    - **If your toString() method is implemented correctly as described below, you will simply need to call it and print the results to the user..**

- **public Sring toString()**
  - ***Brief:***
    - **Gets the String representation of this Team object, which is a neatly formatted table of each Player in the Team on its own line with its position number as shown in the sample output.**
  - ***Returns:***
    - **The String representation of this Team object.**

3. TeamManager

Write a fully-documented class named TeamManager which tests the methods of the Team class and allows the user to manipulate 5 Team objects by performing operations on it.

- **public static final int MAX_TEAMS = 5;**

- **public static void main(String[] args)**
  - ***Brief:***
    - **The main method runs a menu driven application which first creates an empty Team and then prompts the user for a menu command selecting the operation. The required information is then requested from the user based on the selected operation. Following is the list of menu options and their required information:**

- **A) Add Player.** *<Name> <Hits> <Errors> <Position>*
- **G) Get Player stats.** *<Position>*
- **L) Get leader in a stat.** *<Stat>*
- **R) Remove a player.** *<Position>*
- **P) Print all players.** *<Team>*
- **S) Size of team.** *<Team>*
- **T) Select team** *<Index>*
- **C) Clone team** *<From> <To>*
- **E) Team equals** *<Team1> <Team2>*
- **U) Update stat.** *<Name> <Stat> <numHits>*
- **Q) Quit.**

4. FullTeamException

An Exception class which indicates a full roster.

Input Format:

- **Each menu operation is entered on its own line and should be case insensitive (i.e. 'q' and 'Q' are the same).**
- **Check to make sure that the position, if required, is valid. If not, print an error message and return to the menu.**
- **For the Add Player command, if the input information is valid, construct the object accordingly. Otherwise, print an error message and return to the menu.**
- **You may assume that the lengths of the input for the player names are less than 25 characters long, and that the number of hits and errors fits into an int variable.**

Output Format:

- **All lists must be printed in a nice and tabular form as shown in the sample output. You may use C style formatting as shown in the following example. The example below shows two different ways of displaying the name and address at pre-specified positions 21, 26, 19, and 6 spaces wide. If the '-' flag is given, then it will be left-justified (padding will be on the right), else the region is right-justified. The 's' identifier is for strings, the 'd' identifier is for integers. Giving the additional '0' flag pads an integer with additional zeroes in front.**

```
String name = "Doe Jane";
String address = "32 Bayview Dr.";
String city = "Fishers Island, NY";
int zip = 6390;

System.out.println(String.format("%-21s%-26s%19s%06d", name, address, city, zip));
System.out.printf("%-21s%-26s%19s%06d", name, address, city, zip);

Doe Jane             32 Bayview Dr.            Fishers Island, NY 06390
Doe Jane             32 Bayview Dr.            Fishers Island, NY 06390
```

Sample Input/Output:

// Comment in green, input in red, output in black

```
Welcome to TeamManager!

Team 1 is currently selected.

// menu
Please select an option:
A)  Add Player.
G)  Get Player stats.
L)  Get leader in a stat.
R)  Remove a player.
P)  Print all players.
S)  Size of team.
T)  Select team
C)  Clone team
E)  Team equals
U)  Update stat.
Q)  Quit.

Select a menu option: A

Enter the player's name: David Ortiz
Enter the number of hits: 542
Enter the number of errors: 10
Enter the position: 1
Player added: David Ortiz - 542 hits, 10 errors


// menu not shown in the sample input/output
Select a menu option: A

Enter the player's name: Derek Jeter
Enter the number of hits: 83
Enter the number of errors: 20
Enter the position: 1
Player added: Derek Jeter - 83 hits, 20 errors


// menu not shown in the sample input/output
Select a menu option: P

Select team index: 1

Player#   Name               Hits       Errors
-------------------------------------------------
1         Derek Jeter        83         20
2         David Ortiz        542        10


// menu not shown in the sample input/output
Select a menu option: A

Enter the player's name: Albert Pujols
Enter the number of hits: 222
Enter the number of errors: 15
Enter the position: 2
Player added: Albert Pujols - 222 hits, 15 errors


// menu not shown in the sample input/output
Select a menu option: P

Select team index: 1

Player#   Name               Hits       Errors
-------------------------------------------------
1         Derek Jeter        83         20
2         Albert Pujols      222        15
3         David Ortiz        542        10


Select a menu option: S
There are 3 players(s) in the current Team.


// menu not shown in the sample input/output
Select a menu option: R

Enter the position: 1
Player Removed at position 1

Derek Jeter has been removed from the team.


// menu not shown in the sample input/output
Select a menu option: P

Select team index: 1

Player#   Name               Hits       Errors
-------------------------------------------------
1         Albert Pujols      222        15
2         David Ortiz        542        10


// menu not shown in the sample input/output
Select a menu option: A

Enter the player's name: Randy Johnson
Enter the number of hits: 2
Enter the number of errors: 3
Enter the position: 3
Player added: Randy Johnson - 2 hits, 3 errors


// menu not shown in the sample input/output
Select a menu option: P

Select team index: 1

Player#   Name               Hits       Errors
-------------------------------------------------
1         Albert Pujols      222        15
2         David Ortiz        542        10
3         Randy Johnson      2          3


// menu not shown in the sample input/output
Select a menu option: G

Enter the position: 3

Randy Johnson - 2 hits, 3 errors


// menu not shown in the sample input/output
Select a menu option: L

Enter the stat: hits

Leader in hits: David Ortiz - 542 hits, 10 errors


// menu not shown in the sample input/output
Select a menu option: L

Enter the stat: errors

Leader in errors: Randy Johnson - 2 hits, 3 errors


// menu not shown in the sample input/output
Select a menu option: U

Enter the player to update: Albert Pujols
Enter stat to update: errors
Enter the new number of errors: 2

Updated Albert Pujols errors


// menu not shown in the sample input/output
Select a menu option: L

Enter the stat: errors

Leader in errors: Albert Pujols - 222 hits, 2 errors


// menu not shown in the sample input/output
Select a menu option: T

Enter team index to select: 2

Team 2 has been selected.


//Adding players to team 2...
...

// menu not shown in the sample input/output
Select a menu option: P

Select team index: 1

Player#   Name               Hits       Errors
-------------------------------------------------
1         Albert Pujols      222        2
2         David Ortiz        542        10
3         Randy Johnson      2          3


// menu not shown in the sample input/output
Select a menu option: P

Select team index: 2

Player#   Name               Hits       Errors
-------------------------------------------------
1         Miguel Cabrera     374        9
2         Mike Trout         654        11


// menu not shown in the sample input/output
Select a menu option: E

Select first team index: 1
Select second team index: 2

These teams are not equal.


// menu not shown in the sample input/output
Select a menu option: C

Select team to clone from: 2
Select team to clone to: 1

Team 2 has been copied to team 1.


// menu not shown in the sample input/output
Select a menu option: P

Select team index: 1

Player#   Name               Hits       Errors
-------------------------------------------------
1         Miguel Cabrera     374        9
2         Mike Trout         654        11


// menu not shown in the sample input/output
Select a menu option: E

Select first team index: 1
Select second team index: 2

These teams are equal.


//Invalid Input examples:


//Player not found:
// menu not shown in the sample input/output
Select a menu option: U

Enter the player to update: Rick Santorum
Enter stat to update: errors
Enter the new number of errors: 837

Player not found.


//Invalid index:
// menu not shown in the sample input/output
Select a menu option: R

Enter the position: 4
No player at position 4 to remove.


//Invalid index:
// menu not shown in the sample input/output
Select a menu option: A

Enter the player's name: Someone Else
Enter the number of hits: 823
Enter the number of errors: 36
Enter the position: 9
Invalid position for adding the new player.


//Invalid team:
// menu not shown in the sample input/output
Select a menu option: T

Enter team index to select: 7 // 7 > MAX_TEAMS

Invalid index for team.

//Invalid stat:
// menu not shown in the sample input/output
Select a menu option: L

Enter the stat: steals

No such statistic.


// menu not shown in the sample input/output
Select a menu option: Q

Program terminating normally...
```
