Assignment 5

Due time: 03/13/2022, 11:59pm

Total credits: 100, 2 questions

Submission guide:

1. Create a folder and name it with the format FirstName_LastName_Aassigment5 for example Chunyu_Yuan_Assignment5

2. Inside the folder, you should have 4 java files (Player.java, Person_information.java, Club_information.java, TestPlayer.java, StackofInteger.java)

3. compress your file to .zip format and submit it to the blackboard,

4. if you have any question, please send email to cyuan1@gradcenter.cuny.edu

1. continue to design and implement TestPlayer Class (50 credits)

In the assignment4, you have designed the player class and test it using the mlp_players.txt. In the assignment5, please apply hashmap, arraylist and your players object to answer below questions. (You may also use array based on your program designing)

(1) How many players are in each team among the mlp_players.txt? (10 credits)

(2) Display the position distribution in each team. (15 credits)

(In each team, the players are in different positions. What you need to do is to print out the number of players in each position. Here is one example answer: "In BAL: Catcher: 4 players, First Baseman: 5 players……", you can design the output format.)

(3) How many players are in each position among the mlp_players.txt? (10 credits)

It is different from the question 1. The answer is not related to team.

(4) Display the team distribution in each position. (15 credits)

It is like the question 2.

2. continue to design and implement StackofInteger Class (50 credits)

In the class, we discuss and implement the StackofInteger Class. In this assignments, please add below methods to your StackofInteger Class

(1) toCopy(StackofInteger stack), void, given a StackofInteger stack, make the elements and size are the same to stack's

(2) toClear(), void, remove all the value from the StackofInteger stack, empty the StackofInteger stack

(3) getMax(), return int, return the maximum int value of StackofInteger, if there is no value in the stack, print out "no value in the stack" and return the maximum value of Integer

(4) getMin(), return int, return the minimum int value of StackofInteger, if there is no value in the stack, print out "no value in the stack" and return the minimum value of Integer

(5) getDuplicate(), return ArrayList<Integer> that contains all the dulplicate value inside the stack, if there is no duplicate value, print out "no dulplicate" and return an empty ArrayList<Integer>.

Please double-check your methods can work well by yourself. I will use different use cases to test your methods. !!!