COP-2250 Java Programming

Programming Assignment 5

FIU Knight Foundation School of Computing & Information Sciences

For this programming assignment, you will re-implement the java program that you implemented in the fourth programming assignment for storing grades of students, finding the average of grades, the maximum and minimum grade, and printing the name and grade of a student given their ID number.

Program Input Handling

This phase of the program is the same as the previous programming assignment, except this time, you need to do error handling as well. The program first asks the user to enter the information of the first student via keyboard by printing the following message: After

> Welcome to my grade book version 2!
> Please enter the information of the first student using the following format:
> "firstName lastName PID grade".
> Press Enter when you are done.

user enters the requested information (e.g. "Ann Smith 1234567 93"), the program must repetitively ask for information of the next student by printing the following message in a while loop:

> Please enter the information of the next student using the same format.
> If there is no more students, please enter the keyword "DONE".
> Press Enter when you are done.

The program must ask the user to try again (re-enter a student's information) if the input does not meet any of the following criteria:

- firstName is a single word and does not include any white space character.
- lastName is a single word and does not include any white space character.
- PID is a seven digit integer with no leading zeros.
- grade is a non-negative integer that doesn't exceed 100.

Data Structure of the Program

For keeping students' information, you need to define the Student class which includes the following methods and instance fields (You will need to add more methods to the class):

```
public class Student{
    private String firstName;
    private String lastName;
    private int pID;
    private int grade;
    public Student(String givenFirstName, String givenLastName, int givenPID,
        int givenGrade)
    {
        //implement this constructor properly...
    }
    public String getFullName()
    {
        //implement this method properly... returns first + " " + last names.
    }
    public int getGrade()
    {
        //implement this method properly
    }
    public void setGrade(int newGrade)
    {
        //implement this method properly...
    }
}
```

Also, you need to keep the list of all students in an variable of type ArrayList⟨Student⟩ in the main method.

Static Methods of the Program

Other than the main method, your program needs to have the following methods:

```
public static double findAverage(ArrayList⟨Student⟩ students)
{
   //Your code comes here!
}
public static int findMinimum(ArrayList⟨Student⟩ students)
{
   //Your code comes here!
}
public static int findMaximum(ArrayList⟨Student⟩ students){
   //Your code comes here!
}
public static int findGrade(ArrayList⟨Student⟩ students, int givenPID)
{
   //Your code comes here!
}
public static String findFullName(ArrayList⟨Student⟩ students, int givenPID)
{
   //Your code comes here!
   //This method returns the full name (first + " " + last)
}
public static String changeGrade(ArrayList⟨Student⟩ students, int givenPID,
    int newGrade)
{
   //Your code comes here!
   //This method returns the full name (first + " " + last)
}
```

Program Command Handling

The command-handling phase is the same as the previous programming assignment, except this time, you will handle one more command for changing a student's grade.

After the user enters the information of all students one-by-one and enters the keyword "DONE", your program must keep asking for a new command (e.g. print the message "Please enter a new command") and respond to each command properly. Here are the list of all commands that your program must support:

- *average*: Your program must calculate and print the average grade of all students as the response to this command.
- *min*: Your program must calculate and print the minimum grade of all students as the response to this command.

- *max*: Your program must calculate and print the maximum grade of all students as the response to this command.
- *grade XXXXXXX*: Your program must find and print the grade of the student whose PID is given by the command (XXXXXXX is the PID where each X represents a digit).
- *name XXXXXXX*: Your program must find and print the full name of the student whose PID is given by the command (XXXXXXX is the PID where each X represents a digit). Full name is made of first name, followed by a single white space and the last name.
- *change XXXXXXX YY*: Your program must find and update the grade of the student whose PID is given by the command (XXXXXXX is the PID where each X represents a digit). Int this command, YY is the symbol for the new grade.
- *quit*: Your program must stop asking for more commands and quit.

Submissions and Grading Criteria

You need to submit a two files named "GradeBook.java" and "Student.java". Here are the grading criteria:

- Code readability: 5%
- Using comments to explain every line of the program: 5%
- Correctness of Java syntax (no compilation error): 5%
- Proper implementation of Student class: 25%
- Correct implementation of each of the six methods (excl. the main method): 5% each, 30% total.
- Proper usage and naming of variables: 5%
- Proper implementation of the main method including the input-handling phase (when the students' info are stored in an ArrayList⟨Student⟩) and the command-handling phase (when the program parses each command, calls appropriate methods, and prints the results): 25% (15% for input-handling and 10% for command-handling)