

Documentation

Target assessment level

Target assessment level of this work is 3.

Specification

What does the program do?

The program

1. reads data about students from user as a text input
2. allows user to add grades for students
3. allows user to print all given grades and GPAs for all students

Data format

User input

User input consists of strings (student names) and integers (grades). Strings A, G, P and Q are used to control operations. A means adding student, G means adding grades for a student, P means printing the information about all students and Q means quitting the program. "." is used to stop reading grades because it is not an integer. This could also be many other characters. The user can input as many integer grades as they want.

Correctness

Typical test case

The following is a script obtained when running the program:

```
Different operations:
```

```
[Q to quit]
```

```
[A to add a new student]
```

```
[G to add a grade or grades]
```

```
[P to print all the GPAs]
```

```
Choose operation: A
```

```
Please give the name of the student: Alma
```

```
Choose operation: A
```

Please give the name of the student: Bella
Choose operation: A
Please give the name of the student: Cecilia
Choose operation: G
Please give the name of the student: Alma
Please add grades for the student: ['. ' to end writing] 2 17 4 3 6 .
Choose operation: G
Please give the name of the student: Bella
Please add grades for the student: ['. ' to end writing] 1 28 56 3 9 .
Choose operation: G
Please give the name of the student: Alma
Please add grades for the student: ['. ' to end writing] 2 17 .
Choose operation: P
Alma: [2, 17, 4, 3, 6, 2, 17] - GPA: 7.285714285714286
Bella: [1, 28, 56, 3, 9] - GPA: 19.4
Cecilia: [] - GPA: 0.0
Choose operation: Q
Thank you for using GCS. Bye!

Non-trivial algorithms

Selection sort

Data about students is sorted in lexicographic order according to student's name using selection sort using programmer-defined function `sortStudents`.

Searching

Student whose name starts with given name (input from user) is searched using binary search in programmer-defined function `findStudent`. For this the arraylist including all students is sorted using function `sortStudents`.

Arrays

Students are stored into array for sorting and binary search. To convert arraylists, which are used to store list of students in other parts of code, into arrays, is used programmer-defined functions ListToArray and ArrayToList.

Dynamic data structures

- An arraylist is used to read an unspecified amount of data from user input and then store that data of students

Classes

A class is used to store the data of each student. This class is called Student. It stores data about student's name and grades. The only thing taken as parameter for constructor is name, and the grades are added later. The name is used to identify the right student object.