CS 1083

Module 3 Assignment

By Ngoc Phuong Anh Nguyen - 3712361

July 04th 2021

Source Code:

StudentBinarySearch.java:

```
/**
 * Provides a variety of methods for searching a Student object.
 * @author Ngoc Phuong Anh Nguyen - 3712361
*/
public class StudentBinarySearch
{
     /**
      * Constant to signify an unsuccessful search.
      */
     public static final int NOTFOUND = -1;


     /**
      * Searches for a index of a Student object in an array of
Student object, sorted by student ID(ascending order) using the
binary search algorithm.
      * @param stu A sorted array of Student objects, in
ascending order according to their student id values.
      * @param count A counter indicating how many Student
objects are in the array.
      * @param studentID A student ID value.
      * @return The index where a Student object with the given
student id value is located.
      */
     public static int studentBinarySearch(Student[] stu, int
count, int studentID)
     {
          int foundPosition = NOTFOUND;
          int low = 0;
          int high = count - 1;
          int mid;

          while(foundPosition == NOTFOUND && low <= high)
          {
               mid = (low + high) / 2;
               if(stu[mid].getStudentId() == studentID)
```

```
                {
                     foundPosition = mid;
                }
                else if(studentID > stu[mid].getStudentId())
                {
                     low = mid + 1;
                }
                else
                     high = mid - 1;
           }
           return foundPosition;
     }
}
```

StudentSearchTest.java:

```
/**
 * This is a driver program.
 * @author Ngoc Phuong Anh Nguyen - 3712361
 */
public class StudentSearchTest
{
    /**
     * This method creates a randomly generated array of student objects
     * @param baseId A starting point for generating pseudo-random student id values.
     * @param arraySize The size of the desired array.
     * @param studentsNumber The number of array elements to fill in other words, the number of Student objects to be created and inserted into the array
     * @return a randomly generated array of student objects.
     */
    public static Student[] generateStudentArray(int baseId, int arraySize, int studentsNumber)
    {
        Student[] students = new Student[arraySize];
```

```
        String[] surname =
{"Olson","Jackson","Tremblay","Roy","Gagnon","Bouchard","Anderso
n","Cameron","Hamilton","Morrison"};
        String[] givenName =
{"Lucas","Amy","Logan","Jacob","Alex","Mellisa","Jason","Anna","
Emma","Charlotte"};
        int i = 0;


        while(i<studentsNumber)
        {
            students[i] = new Student(
                    baseId + (int)(Math.random()*(80000 - 1 +
1)) + 1,
                    surname[(int)(Math.random()*(surname.length
- 1 + 1))],

givenName[(int)(Math.random()*(givenName.length - 1 + 1))],
                    ((int)(Math.random()*(40-20+1))+20)/10.0
                    );
            int j=0;
            boolean found = false;

            while(!found && j<i)
            {

if(students[i].getGivenNames().equals(students[j].getGivenNames(
)) && students[i].getSurname().equals(students[j].getSurname()))
                {
                    found = true;
                }
                j++;
            }

            if(!found)
            {
                baseId = students[i].getStudentId();
                i++;
            }
        }
```

```
        return students;
    }

    public static void main(String[] args)
    {
        Student[] student = generateStudentArray(3182811,20,10);

        for(int i = 0;i < 10;i++)
        {
            System.out.println("["+i+"]
"+student[i].toString());
        }

        System.out.println("\nSt. Id  Result\n======= =======");

        System.out.println(student[0].getStudentId()+"
"+StudentBinarySearch.studentBinarySearch(student,10,student[0].
getStudentId()));
        System.out.println(student[9].getStudentId()+"
"+StudentBinarySearch.studentBinarySearch(student,10,student[9].
getStudentId()));
        int temp = (int) (Math.random()*10);
        System.out.println(student[temp].getStudentId()+"
"+StudentBinarySearch.studentBinarySearch(student,10,student[tem
p].getStudentId()));

        System.out.println((student[0].getStudentId()-1)+"
"+StudentBinarySearch.studentBinarySearch(student,10,(student[0]
.getStudentId()-1)));
        System.out.println((student[9].getStudentId()+1)+"
"+StudentBinarySearch.studentBinarySearch(student,10,(student[9]
.getStudentId()+1)));
        System.out.println((student[temp].getStudentId()+1)+"
"+StudentBinarySearch.studentBinarySearch(student,10,(student[te
mp].getStudentId()+1)));
    }
}
```

Output:

Case 1:

```
[0] Student[studentId=3221435, surname=Jackson, givenNames=Mellisa, gpa=2.2]
[1] Student[studentId=3239660, surname=Morrison, givenNames=Mellisa, gpa=3.7]
[2] Student[studentId=3251545, surname=Jackson, givenNames=Alex, gpa=2.7]
[3] Student[studentId=3270087, surname=Gagnon, givenNames=Mellisa, gpa=3.1]
[4] Student[studentId=3308371, surname=Anderson, givenNames=Jacob, gpa=4.0]
[5] Student[studentId=3356133, surname=Gagnon, givenNames=Charlotte, gpa=2.7]
[6] Student[studentId=3381399, surname=Jackson, givenNames=Amy, gpa=3.3]
[7] Student[studentId=3422887, surname=Roy, givenNames=Charlotte, gpa=2.7]
[8] Student[studentId=3482478, surname=Cameron, givenNames=Lucas, gpa=2.7]
[9] Student[studentId=3540750, surname=Jackson, givenNames=Lucas, gpa=2.1]

St. Id  Result
======= =======
3221435    0
3540750    9
3356133    5
3221434    -1
3540751    -1
3356134    -1
```

Case 2:

```
[0] Student[studentId=3251189, surname=Gagnon, givenNames=Amy, gpa=2.1]
[1] Student[studentId=3299977, surname=Morrison, givenNames=Jason, gpa=2.8]
[2] Student[studentId=3317386, surname=Roy, givenNames=Lucas, gpa=4.0]
[3] Student[studentId=3368531, surname=Jackson, givenNames=Mellisa, gpa=2.3]
[4] Student[studentId=3393593, surname=Cameron, givenNames=Emma, gpa=3.4]
[5] Student[studentId=3415554, surname=Roy, givenNames=Mellisa, gpa=3.5]
[6] Student[studentId=3465305, surname=Tremblay, givenNames=Charlotte, gpa=3.2]
[7] Student[studentId=3500168, surname=Roy, givenNames=Amy, gpa=2.4]
[8] Student[studentId=3546723, surname=Gagnon, givenNames=Logan, gpa=2.0]
[9] Student[studentId=3605062, surname=Bouchard, givenNames=Lucas, gpa=3.9]

St. Id  Result
======= =======
3251189    0
3605062    9
3299977    1
3251188    -1
3605063    -1
3299978    -1
```