Consider a system for processing student test scores. The following class will be used as part of this system and contains a student's name and the student's answers for a multiple-choice test. The answers are represented as strings of length one with an omitted answer being represented by a string containing a single question mark ("?"). These answers are stored in an ArrayList in which the position of the answer corresponds to the question number on the test (question numbers start at 0). A student's score on the test is computed by comparing the student's answers with the corresponding answers in the answer key for the test. One point is awarded for each correct answer and ¼ of a point is deducted for each incorrect answer. Omitted answers (indicated by "?") do not change the student's score.

(Рассмотрим систему обработки результатов тестов учащихся. Следующий класс будет использоваться как часть этой системы и содержит имя учащегося и ответы учащегося на тест с несколькими вариантами ответов. Ответы представлены в виде строк длины один, а пропущенный ответ представлен строкой, содержащей один вопросительный знак ("?"). Эти ответы хранятся в списке ArrayList, в котором позиция ответа соответствует номеру вопроса в тесте (номера вопросов начинаются с 0). Оценка учащегося за тест вычисляется путем сравнения ответов учащегося с соответствующими ответами в ключе ответов для теста. За каждый правильный ответ начисляется один балл, а за каждый неправильный ответ вычитается ¼ балла. Пропущенные ответы (обозначаются знаком «?») не меняют оценку учащегося.)

public class StudentAnswerSheet

{

private ArrayList answers; // the list of the student's answers

/\*\* @param key the list of correct answers, represented as strings of length one

\*  **Precondition**: key.size() is equal to the number of answers in this answer sheet

\* @return this student's test score (вернуть тест этого учащегося)

\*/

public double getScore(ArrayList key)

{ /\* to be implemented in part (a) \*/ }

/\*\* @return the name of the student

\*/

public String getName()

{ /\* implementation not shown \*/ }

// There may be fields, constructors, and methods that are not shown.

}

The following table shows an example of an answer key, a student's answers, and the corresponding point values that would be awarded for the student's answers. In this example, there are six correct answers, three incorrect answers, and one omitted answer. The student's score is ((6 \* 1) - (3 \* 0.25)) = 5.25 .

(В следующей таблице показан пример ключа ответа, ответов учащегося и соответствующих значений баллов, которые будут начисляться за ответы учащегося. В этом примере есть шесть правильных ответов, три неправильных ответа и один пропущенный ответ. Оценка студента ((6 \* 1) - (3 \* 0,25)) = 5,25.)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер вопроса | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Ключ | “A” | “C” | “D” | “E” | “B” | “C” | “E” | “B” | “B” | “C” |
| Отчет | “A” | “B” | “D” | “E” | “A” | “C” | “?” | “B” | “D” | “C” |
| баллы | 1 | -0.25 | 1 | 1 | -0.25 | 1 | 0 | 1 | -0.25 | 1 |

Write the StudentAnswerSheet method getScore. The parameter passed to method getScore is an ArrayList of strings representing the correct answer key for the test being scored. The method computes and returns a double that represents the score for the student's test answers when compared with the answer key. One point is awarded for each correct answer and ¼ of a point is deducted for each incorrect answer. Omitted answers (indicated by "?") do not change the student's score.

(Напишите метод StudentAnswerSheet getScore. Параметр, передаваемый методу getScore, представляет собой ArrayList строк, представляющих правильный ключ ответа для оцениваемого теста. Метод вычисляет и возвращает двойное число, представляющее оценку тестовых ответов учащегося по сравнению с ключом ответа. За каждый правильный ответ начисляется один балл, а за каждый неправильный ответ вычитается ¼ балла. Пропущенные ответы (обозначаются знаком «?») не меняют оценку учащегося.

Полный метод getScore ниже.)

Complete method getScore below.

/\*\* @param key the list of correct answers, represented as strings of length one

\* **Precondition:** key.size() is equal to the number of answers in this answer sheet

\* @return this student's test score

\*/

public double getScore(ArrayList key)

Consider the following class that represents the test results of a group of students that took a multiple-choice test.

(Рассмотрим следующий класс, который представляет результаты теста группы учащихся, прошедших тест с несколькими вариантами ответов.)

public class TestResults

{

private ArrayList sheets;

/\*\* Precondition: sheets.size() 0 > ;

\* all answer sheets in sheets have the same number of answers

\* @param key the list of correct answers represented as strings of length one

\* Precondition: key.size() is equal to the number of answers

\* in each of the answer sheets in sheets

\* @return the name of the student with the highest score

\*/

public String highestScoringStudent(ArrayList key)

{ /\* to be implemented in part (b) \*/ }

// There may be fields, constructors, and methods that are not shown.

}

Write the TestResults method highestScoringStudent, which returns the name of the student who received the highest score on the test represented by the parameter key. If there is more than one student with the highest score, the name of any one of these highest-scoring students may be returned. You may assume that the size of each answer sheet represented in the ArrayList sheets is equal to the size of the ArrayList key.

Напишите метод HighScoringStudent TestResults, который возвращает имя учащегося, получившего наивысший балл за тест, представленный ключом параметра. Если есть более одного студента с наивысшим баллом, может быть возвращено имя любого из этих студентов с наивысшим баллом. Вы можете предположить, что размер каждого листа ответов, представленного в листах ArrayList, равен размеру ключа ArrayList.

In writing highestScoringStudent, assume that getScore works as specified, regardless of what you wrote in part (a).

(При написании highScoringStudent предположим, что getScore работает так, как указано, независимо от того, что вы написали частично.)

Complete method highestScoringStudent below.

/\*\* Precondition: sheets.size() > 0 ;

\* all answer sheets in sheets have the same number of answers

\* @param key the list of correct answers represented as strings of length one

\* Precondition: key.size() is equal to the number of answers

\* in each of the answer sheets in sheets

\* @return the name of the student with the highest score

\*/

public String highestScoringStudent(ArrayList key)