תרגיל בית TypeScript

|  |  |
| --- | --- |
| **חומר הלימוד לתרגיל:** ירושה ופולימורפיזם | |
| **מספר שאלות:** 3 |

בתרגיל זה נכתוב חלק ממערכת להפקת שאלונים ממוחשבים

שאלה 1 (30%)

כתבו מחלקה מופשטת בשם Question. המחלקה תייצג שאלה. המחלקה תכיל תכונה אחת בשם qText מסוג String שתייצג את טקסט השאלה.

הוסיפו למחלקה בנאי שיקבל את טקסט השאלה ויציב אותו בתכונה, ושיטת set עבור התכונה.

הוסיפו למחלקה שיטת toString שתחזיר את טקסט השאלה.

הוסיפו למחלקה שיטות מופשטות שחתימותיהן –

public abstract String getCorrectAnswer();

public abstract void addCorrectAnswer(String answer);

במערכת שלנו יהיו שני סוגים של שאלות – שאלות פתוחות (שאלות בהן יש תשובה מילולית אחת לשאלה) ושאלות מרובות ברירה (אמריקאיות – שאלות בהן התשובה הנכונה היא אחת מבין כמה אופציות).

כתבו מחלקה בשם ShortAnswerQuestion היורשת מהמחלקה Question, המייצגת שאלה פתוחה. המחלקה תכיל תכונה בשם answer מסוג String המייצגת את התשובה לשאלה.

הוסיפו למחלקה בנאי המקבל את טקסט השאלה ואת טקסט התשובה ומציב אותם בתכונות.

דרסו במחלקה את השיטות המופשטות. השיטה getCorrectAnswer תחזיר את טקסט התשובה הנכונה, והשיטה addCorrectAnswer תציב בתכונת התשובה את הפרמטר שהתקבל בשיטה.

כתבו מחלקה בשם MultipleChoiceQuestion היורשת מהמחלקה Question, המייצגת שאלה רבת ברירה. ניתן להניח שלשאלה רבת ברירה יהיו לכל היותר 6 תשובות אפשריות (אבל יכולות להיות פחות). המחלקה תכיל את התכונות הבאות - מערך של מחרוזות בשם answers שייצג את התשובות האפשריות לשאלה, משתנה מסוג int בשם numberOfAnswers שייצג את מספר התשובות שיש לשאלה, ומשתנה מסוג int בשם correctAnswerIndex שייצג את האינדקס במערך של התשובה הנכונה (ניתן להניח שלכל שאלה יש תשובה נכונה אחת **בלבד**).

הוסיפו למחלקה בנאי שמקבל את טקסט השאלה וטקסט **התשובה הנכונה**. הבנאי יציב את התשובה בתא הראשון במערך.

הוסיפו למחלקה שיטת toString שתחזיר כמחרוזת את השאלה במבנה הבא –

טקסט השאלה יופיע בשורה אחת ואחריו כל התשובות האפשריות, כל תשובה בשורה משלה.

דרסו במחלקה את השיטות המופשטות. השיטה getCorrectAnswer תחזיר את טקסט התשובה הנכונה לשאלה. השיטה addCorrectAnswer **תוסיף** לתא הפנוי הבא של מערך התשובות את התשובה שהועברה כפרמטר ותעדכן את אינדקס התשובה הנכונה להיות תשובה זו. שימו לב – אם אין מקום במערך התשובות, השיטה לא תעשה דבר.

הוסיפו במחלקה שיטה שחתימתה:

public void addAnswer(String answer)

השיטה תקבל כפרמטר תשובה (לא נכונה) ותוסיף אותה למערך התשובות של השאלה. אם אין מקום במערך התשובות, השיטה לא תעשה דבר.

שאלה 2 (30%)

כתבו מחלקה בשם QuestionsCatalog שתייצג מאגר שאלות. המחלקה תכיל תכונה בשם questions מסוג Question[] (מערך של אובייקטים מסוג Question). ניתן להניח שגודל המערך הוא 20 (הגדירו כקבוע במחלקה). כמו כן תכיל המחלקה תכונה מסוג int שתשמור כמה שאלות יש כרגע במאגר.

בנוסף הגדירו את הקבועים הסטטיים הבאים במחלקה –

public static final int SHORT = 1, MULTIPLE = 2, BOTH = 3;

הוסיפו למחלקה בנאי ריק שיצור את המערך ויציב במספר השאלות 0.

הוסיפו למחלקה את השיטות הבאות:

public void addQuestion(Question q) - השיטה תקבל אובייקט של שאלה ותוסיף אותו למקום הפנוי הבא במערך השאלות. אם אין מקום במערך, השיטה לא תעשה דבר. (מותר להשתמש ב-aliasing).

public Question[] generateQuestionnaire(int num, int type) - השיטה תיצור שאלון של שאלות לפי הפרמטרים – הפרמטר num ייצג את מספר השאלות שיכיל השאלון, והפרמטר type יכיל את סוג השאלות. ערכי type יהיו –

SHORT – רק שאלות פתוחות

MULTIPLE – רק שאלות רבות ברירה.

BOTH – שאלות משני הסוגים.

השיטה תבחר בצורה רנדומלית num שאלות מהסוג type מתוך המערך questions, ותחזיר מערך של השאלות שנבחרו.

שימו לב, שאלה לא יכולה להופיע פעמיים באותו שאלון. אם אין מספיק שאלות מסוג type כדי להגיע ל-num שאלות, השיטה תחזיר רק את כמות השאלות שהיא מצאה (בכל מקרה השיטה תחזיר מערך בגודל מספר השאלות שהיא מצאה, כלומר לא יהיו תאים ריקים במערך).

שאלה 3 (40%)

כתבו מחלקה בשם Questionnaire שתייצג שאלון. המחלקה תייצר שאלון חדש, ותאפשר למשתמש להבחן על השאלון. המשתמש יוכל לענות על השאלון, שאלה אחר שאלה, ולבסוף לקבל את התוצאות. שימו לב, אתם יכולים להחליט על מימוש והמבנה הפנימי של המחלקה בעצמכם, אבל עליכם לכתוב את כל השיטות הציבוריות המפורטות כאן, ורק אותן (אין להוסיף שיטות ציבוריות אחרות או לשנות את החתימות של השיטות המפורטות).

כתבו במחלקה בנאי שחתימתו –

public Questionnaire(int type, int num, QuestionsCatalog cat) – הבנאי יקבל את סוג השאלות שיופיעו בשאלון, מספר השאלות בו ואובייקט מסוג QuestionsCatalog. הבנאי יצור שאלון חדש עם שאלות מהסוג ובכמות המפורטים בחתימת הבנאי. שימו לב שאתם צריכים לדאוג לזה שהאובייט cat יכיל שאלות שמהן יורכב השאלון.

הוסיפו למחלקה את השיטות הבאות:

public boolean hasNext() – השיטה תחזיר true אם יש עוד שאלות שלא נשאלו בשאלון, ו-false אחרת.

public Question getNext() – השיטה תחזיר את השאלה הבאה בשאלון.

public void checkAnswer(String answer) – השיטה תקבל כפרמטר מחרוזת המייצגת את תשובת המשתמש לשאלה הנוכחית.

public int getTotalQuestions() – השיטה תחזיר את כמות השאלות בשאלון.

public int getCorrectAnswers() – השיטה תחזיר את כמות השאלות שנענו נכון בשאלון.

תוכלו להיעזר בטסטר הבא:

function main() {

cat: QuestionsCatalog = new QuestionsCatalog();

q1: Question = new ShortAnswerQuestion("How much legs does the spider have?", "8");

Cat.addQuestion(q1);

q2: Question = new MultipleChoiceQuestion("Where is the sun rise?", "East");

q2.addAnswer("West");

q2.addAnswer("North");

q2.addAnswer("South");

cat.addQuestion(q2);

// add more questions to the catalog

Qnr: Questionnaire = new Questionnaire(5, Catalog.BOTH);

Console.log("Welcome to our questionnaire, its starts now!");

while(qnr.hasNext()) {

q: Question = qnr.getNext();

console.log(q);

console.log("Your answer: ");

answer: string = scan.nextLine();

qnr.checkAnswer(answer);

}

Console.log("Thank you for participating in our test");

Correct: number = qnr.getCorrectAnswers();

Total: number = qnr.getTotalQuestions();

Console.log("You've answered " + correct + " correct answers out of " +

total + " questions");

}

**הנחיות הגשה**

1. עליכם להגיש את הקבצים הבאים בלבד. אין להגיש קבצים נוספים:

Question.ts, ShortAnswerQuestion.ts, MultipleChoiceQuestion.ts, QuestionsCatalog.ts, Questionnaire.ts

1. ארזו את כל הקבצים בקובץ zip או rar יחיד וקראו לו Ex1.zip או Ex1.rar.
2. אין להגיש בזוגות.