

הנחיות הגשה לתרגילים 3-1

1. העבודה על התרגילים היא בזוגות, ניתן להגיש עם בן זוג מקבוצה אחרת של אותו מרצה בלבד.
2. את התרגילים יש להגיש במודל בכניסה המתאימה שתפורסם להגשת התרגיל.
3. יש להגיש קובץ זיפ עם מחלקות הפרוייקט: כל הקוד יהיה כתוב ב- package ששמו כמספרי תעודות הזהות של המגשים, למשל id12345678_id9989899.
4. יש להגיש את הקבצים הבאים:
(א) קובץ **readme** – חובה!
כולל את פרטי המגשים (שם ות"ז) מספר התרגיל והנחיות מיוחדות לבדיקה אם ישנן.
(ב) קובץ **Known Issues** – חובה! בקובץ תיאור באגים קיימים, פירוט בהמשך.
5. בפתרון כל חלק יש להשתמש אך ורק בנושאים שנלמדו בכיתה ומוגדרים בראש החלק, אין להשתמש בנושאים מתקדמים.

קובץ Known Issues

בקובץ זה אתם מתעדים את הבעיות שידועות לכם בתרגיל ולא הספקתם לטפל בהן או שאינכם יודעים כיצד: באגים, פיצ'רים שעדיין לא טיפלתם בהם וכו'. שימו לב: לא ניתן להגיש במודל תרגיל לאחר מועד ההגשה! יתכן בהחלט שההגשה שלכם טובה אבל לא מושלמת, תיעוד של ה-Known Issues הינו כלי מקובל בתעשייה על-מנת לתעד את הנקודות שהפיתוח מודע אליהן. ההתייחסות לבעיות שתיעדתם בקובץ ה-Known Issues תהיה סלחנית יותר לעומת בעיות שיימצאו בבדיקה ולא תועדו. שימו לב: חובה להגיש קובץ Known Issues גם אם לא קיימים Known Issues!

הנחיות כלליות לכתיבת המערכת

1. הגדלה של מערכים תהיה כל פעם פי 2. לכן יש צורך לתחזק משתנה לגודל הלוגי ולגודל הפיזי.
2. צריך "מחלקה מנהלת" שהיא המחלקה שמחזיקה את כל המידע ורק מולה ה-main יעבוד.
3. לא יהיה קלט במחלקות וגם לא במחלקה המנהלת. ה-main הוא זה שקולט את המידע ושולח ערכים כפרמטרים לפונקציות. באופן זה גם ניתן להכניס למערכת ערכים התחלתיים ב-main במקום להקלידם בכל הרצה מחדש.
4. עליכם להחליט כיצד התפריט יראה: מה הקלט, מה תת-התפריטים וכו'. הגיוני כי כל תרגיל יהיה שונה, אתם נבדקים על איכות הקוד, ולא על פרשנות מינורית כזו או אחרת של איך בדיוק הדברים צריכים להראות.
5. בדיקות תקינות:
a. אין צורך להמציא הגבלות על הקלטים, אלא להתייחס רק להגבלות הנתונות בגוף השאלה. יחד עם זאת, להקפיד שמתודות ה-set תמיד יחזירו boolean כתשתית לבדיקת תקינות עתידית.

- b. ניתן להניח שהקלט המתקבל הוא מהטיפוס המבוקש, כלומר אם הקלט הצפוי הוא מספר, ניתן להניח שאכן הוקלד מספר. אך אם יש בטקסט הגבלות למשל לטווח ערכים אפשרי, אז יש לוודא שהמספר שהוקלד בטווח המתאים.
- c. אתם יכולים להתחיל לעבוד כל החלק הראשון גם כל עוד לא למדתם חריגות (exceptions), ובינתיים ניתן להניח שפרמטרים המועברים לקונסטרקטור תמיד תקינים, ואין צורך לטפל בקלט שגוי בקונסטרקטור. הסיבה לכך היא שבפרק של exception נראה דרך אלגנטית לטפל בערכים שגויים גם בקונסטרקטור.

מידע כללי

התרגיל מתחלק ל- 3 חלקים, וזהו פרויקט מתמשך המתפתח לאורך הסמסטר.

חלק 1: מבוסס על הנושאים הבאים בלבד: הורשה, פולימורפיזם,**המחלקה Object וחריגות****תיאור המערכת – מערכת המייצרת מבחנים באופן אוטומטי**

בשאלה זו נכתוב מערכת המייצרת מבחנים הכוללים שאלות פתוחות ושאלות רב-ברירה באופן אוטומטי. לצורך כך עלינו לייצר מאגר של שאלות ותשובות במערכת.

לכל שאלה יהיה מספר מזהה סידורי אוטומטי וטקסט השאלה. חלק מהשאלות הן שאלות פתוחות ועבורן נשמור בנוסף את התשובה המבוקשת. חלק מהשאלות הן אמריקאיות ועבורן נשמור את בנוסף לטקסט השאלה את אוסף תשובות, ועבור כל תשובה יש לשמור האם התשובה נכונה או לא. ניתן להניח שלכל שאלה אמריקאית יש מקסימום 10 תשובות אפשריות.

לא ניתן להוסיף למערכת שאלה שכבר קיימת, ושאלות נחשבות לזהות במידה וטקסט השאלה שלהן זהה. יתכן והטקסט של שאלה פתוחה ושאלה אמריקאית יהיה זהה ואז הם נחשבות שתי שאלות שונות

המערכת תאפשר את הפעולות הבאות בתפריט:

1. הצגת כל השאלות המערכת והתשובות שהוזנו לשאלות. אם זו שאלה אמריקאית, ליד כל תשובה תהיה אינדיקציה האם זוהי תשובה נכונה או לא.
2. הוספת שאלה ותשובה. במידה והשאלה היא אמריקאית, יש לאפשר להוסיף אוסף תשובות ואינדיקציה נכונות לכל תשובה.
3. עדכון נוסח של שאלה קיימת.
4. עדכון נוסח של תשובה קיימת (לשים לב לטפל באופן ידידותי ונוח בתשובה של שאלה אמריקאית).
5. מחיקת תשובה לשאלה.
6. יצירת מבחן באופן ידני:

- התוכנית תשאל את המשתמש כמה שאלות הוא רוצה שיהיו במבחן.
- התוכנית תשאל את המשתמש איזה שאלות הוא רוצה שיהיו במבחן, ובמידה והשאלה אמריקאית, אילו תשובות יופיעו עבורה.
- תוצר סעיף זו הוא הצגת השאלות במבחן. במידה והשאלה אמריקאית, בנוסף לתשובות שנבחרו, יש להוסיף תמיד את שתי התשובות הבאות: "אף תשובה אינה נכונה" ו"יותר מתשובה אחת נכונה".
- בנוסף יוצג שאלון הבחינה כך שלאחר כל שאלה תוצג התשובה.
- 7. יצירת מבחן באופן אוטומטי:
- התוכנית תשאל את המשתמש כמה שאלות הוא רוצה שיהיו במבחן.

- התוכנית תבחר באופן אקראי שאלות כאשר עבור שאלות אמריקאיות יבחרו 4 תשובות עבור כל שאלה. באפשרות זו לא ניתן שתייה יותר מתשובה אחת נכונה לשאלה, אך יתכן ואף תשובה לא תהייה נכונה.
 - תוצר סעיף זו הוא הצגת השאלות במבחן. במידה והשאלה אמריקאית, בנוסף לתשובות שנבחרו, יש להוסיף תמיד את שתי התשובות הבאות: "אף תשובה אינה נכונה" ו"יותר מתשובה אחת נכונה".
 - בנוסף יוצג שאלון הבחינה כך שלאחר כל שאלה תוצג התשובה.
 - השאלות יצגו באופן ממין (בעזרת מימוש מיון כלשהו שלמדתם, אין להשתמש ב- Arrays.sort או מיון מוכן) בסדר מילוני לפי הטקסט בשאלה
8. יציאה מהתפריט

ובנוסף..

המערכת הנוכחית תומכת ביצירת שאלות ותשובות למבחן במקצוע אחד בלבד. בקובץ ה- Program אשר יכיל את פונקציית ה- main, רישמו בהערה אילו שינויים ו/או תוספות ידרשו בפרוייקט על-מנת לתמוך במספר רב של מקצועות.

כמו כן, לפני התפריט יש להוסיף hard-coded לפחות 5 שאלות עם אוסף תשובות וזאת ל- מנת להקל בבדיקות שלא יהיה צריך כל פעם להקליד הכל מחדש.

עבור כל מחלקה דרוסו את המתודות toString ו- equals.

חלק 2: מבוסס על הנושאים הבאים בלבד: ממשקים, קבצים ו-

generics, אופים

עדכנו את המערכת באופן הבא:

1. בנוסף להצגת המבחנים (הידיניים או האוטומטיים) ופתרונותיהם למסך, יש לשמור את התוכן לקבצי טקסט.

שימו לב: שם כל קובץ מבחן שיווצר יהיה בפורמט הבא: exam_yyyy_mm_yy בהתאם לזמן בו נוצר המבחן. שם קובץ התשובות יהיה זהה פרט לכך שבמקום המילה exam בשם הקובץ תופיע המילה solution. חלק התאריך בשם קובץ המבחן וקובץ הפתרון צריך להיות עם ערכים זהים.

2. עם היציאה מהתוכנית, ישמרו לקובץ בינארי כל שאלות המבחנים והתשובות שהוזנו במהלך ריצת התוכנית. עם תחילת התוכנית, יטענו לזיכרון נתוני השאלות והתשובות, כך שבהרצה הבאה כבר יהיה מאגר שאלות מוכן.
3. הוסיפו אפשרו השוואה בין שאלות עפ"י אורך התווים הכולל בתשובה (עבור שאלה אמריקאית זה סה"כ התווים בכל האפשרויות), ודאגו בעת יצירת מבחן למיין את שאלות המבחן כך שהשאלה עם התשובה הקצרה ביותר תהיה ראשונה והשאלה עם התשובה הארוכה ביותר תהיה אחרונה.
4. יש להוסיף לתפריט אופציה של יצירת מבחן חדש כהעתק של מבחן שכבר קיים במערכת.
5. כתבו את המחלקה הגנרית עבור מבנה הנתונים Set. מבנה נתונים זה מכיל ערכים עפ"י סדר הכנסתם, אך אינו מוסיף ערך שכבר קיים (זהו פשוט אוסף ללא כפילויות) שימו לב: קיימת כבר מחלקה קיימת כזו, אך פה עליכם לכתוב אותה בעצמכם.
6. החזיקו את נתוני כל התשובות לשאלה אמריקאית בתוך מחלקת ה- Set שכתבתם. ניסיון הוספת אובייקט מטיפוס שאינו מורשה תגרור שגיאת קומפילציה מתאימה.
7. יש להמיר את השימוש של כל המערכים ב- Vector או ב- ArrayList. לא לשכוח להוריד את הקוד המיותר בעקבות כך.
8. כתבו ממשק אחד לאוסף הפעולות של ה- main כך שאם מישהו בהמשך ירצה לכתוב את התפריט באופן שונה, יהיה לו ממשק המרכז את כל הפעולות שעליו לממש.

חלק 3 (פרוייקט הסיום): GUI ו- MVC

ממשו ממשק משתמש גרפי שיתמוך בכל הפעולות של שני החלקים הראשונים, כלומר בקליטת הנתונים ובהצגת התוצאות באופן גרפי. המימוש צריך להיות מבוסס על מודל ה- MVC.