

# תוכנה 1

## תרגיל מספר 10

### GUI, enum, hashCode and equals

#### הנחיות כלליות:

קראו בעיון את קובץ נהלי הגשת התרגילים אשר נמצא באתר הקורס.

- הגשת התרגיל תעשה במערכת ה-moodle בלבד (<http://moodle.tau.ac.il/>).
- יש להגיש קובץ zip יחיד הנושא את שם המשתמש ומספר התרגיל (לדוגמא, עבור המשתמש aviv1 יקרא הקובץ aviv1\_hw10.zip). קובץ ה-zip יכיל:
  - א קובץ פרטים אישיים בשם details.txt המכיל את שמכם ומספר ת.ז.
  - ב כל מה שנמצא תחת ה-src בשלד הקוד שקיבלתם.

## חלק א': חידות ג'אוה (40%)

1. (25 נקודות) בשאלה זו נשלים קוד ג'אוה, תוך תרגול של הבנת הקוד וחזרה על עקרונות שנלמדו בכיתה בנושאים שונים. באתר הקורס נתונה לכם החבילה riddles, ותחתיה המחלקה: Riddle. עליכם להשלים את המחלקות A, B, C מבלי לשנות את Riddle כלל, ומבלי להוסיף קבצים אחרים, כך ש:

1. כל הקוד יעבור קומפילציה ללא שגיאות וללא אזהרות.
2. אם נריץ את התכנית, היא תמיד תסיים את הריצה ללא שגיאות ותדפיס success!

הערות:

- בקבצים מצוין בדיוק היכן מותר לכם לשנות את הקוד
- מומלץ, בהינתן שגיאת קומפילציה, לחשוב תחילה בעצמכם כיצד לפתור אותה, לפני שתיעזרו בהצעות של eclipse. אם אתם משתמשים בהצעות אלה, היזהרו לא לשנות בטעות את Riddle, משום ששינוי זה עשוי לפגוע בנכונות הפתרון שלכם.
- העזרו במצגות התרגולים כיצד לכתוב את הפונקציות hashCode ו-equals, וחשבו אילו שדות יש לקחת בחשבון בהשוואה.

2. (25 נקודות) בשאלה זו נשלים קוד ג'אוה. באתר הקורס נתונה לכם החבילה enumRiddles, ותחתיה שתי מחלקות TLightTest, DayTest:

- . עליכם להשלים את המחלקות מבלי לשנות את פונקציות ה-main כלל, ומבלי להוסיף קבצים אחרים, כך ש:
- כל הקוד יעבור קומפילציה ללא שגיאות וללא אזהרות.
- אם נריץ את התכנית, הן תמיד תסיים את הריצה ללא שגיאות וידפיסו (ללא ירידות שורה מיותרות בין ההדפסות):

TLightTest:

RED: 30 seconds, next is GREEN  
AMBER: 10 seconds, next is RED  
GREEN: 30 seconds, next is AMBER

DayTest:

MONDAY (1), next is TUESDAY  
TUESDAY (2), next is WEDNESDAY  
WEDNESDAY (3), next is THURSDAY  
THURSDAY (4), next is FRIDAY  
FRIDAY (5), next is SATURDAY  
SATURDAY (6), next is SUNDAY  
SUNDAY (7), next is MONDAY

הערות:

- אין מגבלה על הקוד של המחלקות, כל עוד הוא קוד ג'אווה תקין, והוא מממש את `enum`.
- ניתן וצריך להשתמש במתודות אבסטרקטיות ולממש באופן שונה עבור כל ערך

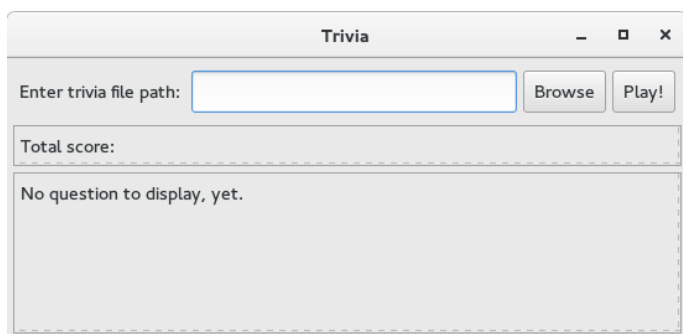
## חלק ב': משחק טריוויה - GUI (50%)

בתרגיל זה נכתוב ממשק משתמש גרפי אשר מתפקד כמשחק טריוויה בסיסי. נתרגל עבודה עם רכיבי GUI שונים והבנת קוד נתון - שלדי המחלקות והמחלקות אשר מופיעים באתר. אינכם נדרשים לבצע פעולות של עיצוב ה-GUI, אלא רק של תפעול האפליקציה לפי מהלך המשחק.

שימו לב, חלק מן החומר הדרוש לפתרון התרגיל יילמד בקורס רק בשבוע הבא, אבל כבר בשלב זה ניתן לממש את החלק הנוגע לטעינת שאלות טריוויה מקובץ ולבחירת שאלת טריוויה אקראית.

### תיאור השימוש בממשק הגרפי

כדי לפתוח את הממשק הגרפי יש להריץ את התכנית TriviaMain. כעת, ייפתח חלון הראשי של התכנית.

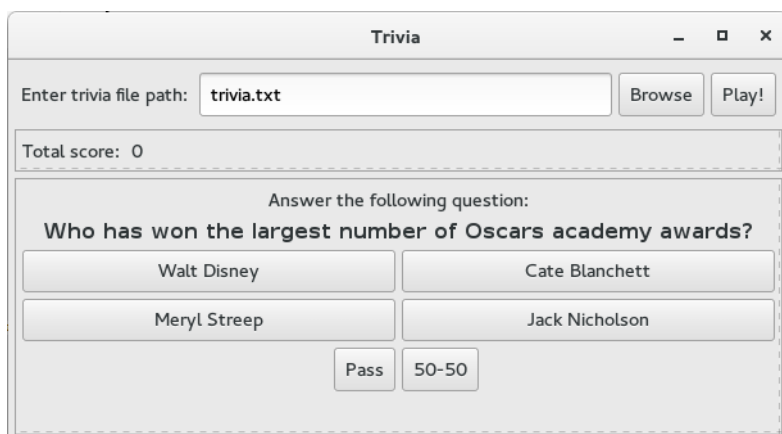


ניתן להקליד מסלול לקובץ בשדה הנתון, או ללחוץ על הכפתור Browse. במקרה הזה, ייפתח חלון לבחירת קובץ.

קבצי טריוויה הם קבצי טקסט בפורמט הבא: כל שורה מכילה שאלה ולאחריה 4 תשובות אפשריות, כאשר הראשונה מביניהן היא נכונה והיתר שגויות. בין השאלה לתשובות ובין התשובות מפריד טאב בודד. לדוגמא,

What is an apple?	Fruit	Vegetable	Fungus	Meat		
When was the Beijing Summer Olympic Games?			2008	2014	2002	2000

לאחר בחירת מסלול הקובץ הרצוי ניתן ללחוץ על Play! בחלון הראשי. אם הקובץ תקין, הוא ייטען ע"י האפליקציה ותוצג למשתמש שאלה אקראית מתוכו ואת ארבע התשובות האפשריות עבורה בסדר אקראי. בשלב זה הניקוד הכולל של המשתמש הוא 0.



המשתמש יכול לבחור באחת התשובות או להשתמש באחד משני גלגלי הצלה: דילוג על השאלה (Pass) ו-50-50.

### חוקי המשחק:

- תשובה נכונה = +3 נקודות
- תשובה שגויה = -2 נקודות

- המשחק מסתיים בעקבות אחד מהאירועים הבאים:
  - 3 תשובות שגויות ברצף (פסילה)
  - נגמרו השאלות במאגר
- גלגלי ההצלה זמינים לשימוש, מתחילת המשחק, ובכל שאלה מחדש תחת התנאים הבאים:
  - השימוש הראשון בכל גלגל הצלה (בנפרד) אפשרי ללא הגבלה או תלות במאזן הניקוד של השחקן
  - לאחר שימוש ראשון בגלגל הצלה מסוים, כל עוד מאזן הניקוד של השחקן קטן או שווה ל-0 לא ניתן לחזור ולהשתמש בו בשאלות נוספות.
  - לדוגמא, לאחר שימוש ראשון בכפתור 50-50, אם הניקוד הוא 2- הגלגל לא זמין לשימוש (לא ניתן ללחוץ עליו). ברגע שהשחקן יקבל ניקוד גדול מ-0 הגלגל יחזור לשימוש ויהיה זמין ללחיצה.
  - לאחר שהשחקן צובר נקודות ומאזנו חיובי ניתן לפתוח את השימוש בכל אחד מגלגלי ההצלה שוב
  - ניקוד שימוש בכל אחד מגלגלי הצלה:
    - חינם בפעם הראשונה
    - כל שימוש נוסף מעבר לפעם הראשונה = -1 נקודה
- כמו כן, תוצג למשתמש שאלה אקראית **חדשה** (המשתמש לעולם לא יקבל את אותה שאלה פעמיים באותו משחק). המשחק כאמור יסתיים לאחר 3 תשובות שגויות ברצף או כאשר סיימנו לעבור על כל השאלות בקובץ. בשלב זה תוצג למשתמש הודעה מסכמת.



## תכנית לדוגמא

נמצאת באתר הקורס. במקרה של ספק לגבי אופן פעולת המשחק במצב מסויים, ניתן להיעזר בדוגמא זו. יש להוריד את הקובץ Trivia.jar ולשמור אותו בתיקיה כלשהי במחשב. באותה תיקיה, שמרו עותק של swt.jar המתאים למערכת ההפעלה שלכם. ניתן להוריד את ה-jar מ-<http://www.eclipse.org/swt>. מומלץ להוריד ע"י לחיצה על Releases > Stable > more... , ולחיצה על השורה המתאימה למע' ההפעלה שלכם. קובץ הזיפ שתורידו יכול בתוכו את swt.jar, ויש להוציא אותו מן הזיפ ולשמור באותה תיקיה יחד עם Trivia.jar. כעת, כדי להריץ את התכנית לדוגמא מ-command line היכנסו לאותה תיקיה (ב-cmd למשתמשי Windows, terminal למשתמשי Linux וכו'), והקישו את הפקודה

```
java -jar Trivia.jar
```

(לחלופין, בחלק ממערכות ההפעלה, לחיצה כפולה על קובץ ה-jar תפעיל אותו) אם פעלתם לפי ההנחיות, ו- java 8 מותקנת על המחשב, יפתח חלון עם משחק הטריוויה.

## מה עליכם לעשות

באתר הקורס נתונים לכם שלדים של מחלקות ה-GUI בחבילה il.ac.tau.cs.sw1.trivia. הוסיפו ושנו קוד לפי הצורך. יש להוסיף ל-build path של הפרויקט את swt.jar המתאים למערכת ההפעלה. באתר הקורס גם נתון קובץ טריוויה תקין לדוגמא, trivia.txt.

## פירוט

**51 נק'** הכפתור Browse: הוסיפו מאזין לכפתור זה כך שלחיצה עליו תפתח חלון לבחירת קובץ, וכך שאם נבחר קובץ המסלול אליו יתעדכן בשדה הטקסט המתאים. היעזרו במתודה `GUIUtils.getFilePathFromFileDialog` הנתונה לכם.

**201 נק'** הכפתור Play!: הוסיפו מאזין לכפתור כך שלחיצה עליו תגרום לטעינת השאלות והתשובות מהקובץ הנתון. ניתן להניח שפורמט הקובץ תקין.

שמרו את השאלות והתשובות במבנה נתונים לבחירתכם. ניתן להוסיף מחלקות כרצונכם. שימו לב, המזהה הייחודי של שאלה הוא הטקסט של השאלה בצירוף קבוצת התשובות שלה. כלומר, ייתכנו שתי שאלות עם אותו טקסט, אך עם קבוצת תשובות שונה (ראו דוגמא ב-`trivia.txt` שורות 6-7).

לבסוף, הציגו שאלה אקראית מן הקובץ ואת התשובות שלה בסדר אקראי. כדי ליצור את הכפתורים היעזרו במתודה `updateQuestionPanel`. ניתן להיעזר במתודה `nextInt` של המחלקה `java.util.Random` כדי לקבל מספר אקראי, ו־`Collections.shuffle` כדי לסדר את רשימת התשובות בסדר אקראי.

אתחלו את הניקוד להיות 0, ושדות אחרים לפי הצורך.

**151 נק'** כפתורי התשובות: הוסיפו מאזינים לכפתורים האלה כך שלחיצה על הכפתור הנכון תגדיל את הניקוד ב-3 ולחיצה על כפתור של תשובה שגויה תקטין את הניקוד ב-1 ותגדיל ב-1 את מספר התשובות השגויות.

בדומה לכפתור ה-`Play!`, יש להציג את השאלה הבאה. נסו לשתף קוד בין מאזיני הכפתורים. במידה והמשחק נגמר (התקבלו 3 תשובות שגויות ברצף או סיימנו את כל השאלות) יש להציג למשתמש הודעה מסכמת. לשם כך, היעזרו במתודה `GUIUtils.showInfoDialog`.

**101 נק'** כפתור Pass: הוסיפו לכפתור זה מאזין המתקדם לשאלה הבאה מבלי לעדכן את מספר השאלות שנענו (המצוין בהודעה המסכמת). זיכרו לעדכן את מצב הכפתור והניקוד בהתאם לחוקי המשחק. שוב, נסו לשתף קוד עם המאזינים לכפתור `Play!` וכפתורי התשובות.

**101 נק'** כפתור 50-50: הוסיפו לכפתור זה מאזין אשר מנטרל שני כפתורי תשובות לא נכונות ומשאיר רק כפתור עם תשובה נכונה ואחד עם תשובה לא נכונה באופן אקראי. זיכרו לעדכן את מצב הכפתור והניקוד בהתאם לחוקי המשחק. שוב, נסו לשתף קוד עם המאזינים לכפתור `Play!` וכפתורי התשובות.

## הערות

- לצורך פתרון התרגיל, חשובה יכולת קריאה והבנה של הקוד הנתון, תוך זיהוי החלקים הרלוונטיים לקוד שתרצו להוסיף. הקוד הנתון מתועד ברובו, אך ייתכן שתצטרכו להיעזר גם בתיעוד של `SWT`, בתיעוד של `java` ובאינטרנט כדי להבין חלקי קוד מסוימים.
- בחלק זה של התרגיל אתם רשאים לשנות את המחלקות והמתודות הנתונות לפי הצורך, מלבד פונקציה ה-`main` הראשית במחלקה `il.ac.tau.cs.sw1.trivia.TriviaMain`.
- לפני תחילת פתרון התרגיל, מומלץ לעבור על כל הקוד, ובפרט על המחלקה `GUIUtils`. ייתכן שתמצאו שירותים שיעזרו לכם בפתרון התרגיל (מומלץ אבל אין חובה להשתמש בהם).
- בדקו את עצמכם ע"י משחק בתכנית.
- יש להגיש את כל המחלקות (קבצי `java`), גם אם לא שיניתם אותן. אין צורך להגיש את `swt.jar`.

בהצלחה !