

תוכנה 1 – חורף 2020/2021

תרגיל מספר 10

GUI, enum, hashCode and equals

הנחיות כלליות:

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

- הגשת התרגיל תיעשה במערכת ה-moodle (<http://moodle.tau.ac.il/>), וניתן להגיש בנוסף להגשה במודל גם בגיט.
- יש להגיש קובץ zip יחיד הנושא את שם המשתמש ומספר התרגיל (לדוגמא, עבור המשתמש stav1 יקרא הקובץ stav1_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. כלומר zip לsrc שלכם כולה, כולל כל הקבצים שנמצאים שם. אם נפתח את קובץ ה zip שתגישו יהיו שם קובץ ה details ותיקיית ה src לא מכווצת.
- להגשה בגיט צרו את repository שלכם מתוך הקישור הבא: <https://classroom.github.com/a/yeR82C4G>

חלק א': חידות ג'אוה (50%)

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. בכל מקום אחר שלא מצויין שניתן לשנות את הקוד – אסור לשנות אותו.
- 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. מומלץ להשתמש גם ב-Debugger כדי להבין אילו פונקציות נקראות בכל שלב. יכולים להיות מספר פתרונות אפשריים לחידות.

- פיתרון החידה יבדק רק עם כתב החידה הנוכחי, אך ערכי המספרים יכולים להשתנות ולכן הפיתרונות שלכם לא אמורים קוד שמתמש במספרים ספציפיים בצורה `hard coded`.

2. (25 נקודות) בשאלה זו נשלים קוד ג'אווה. 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 התכניות, 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:

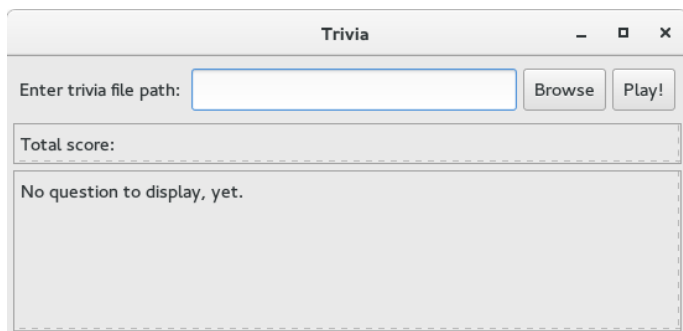
- `TLightTest` תדפיס 3 שורות בלבד ו-`DayTetsi` תדפיס 7 שורות בלבד, ללא ירידות שורה מיותרות בסוף.
- אין מגבלה על הקוד של המחלקות, כל עוד הוא קוד ג'אווה תקין, והוא מממש את `enum`. הדבר היחידי שאסור לשנות אלו מתודות ה-`main` של המחלקות.
- ניתן וצריך (אך לא חובה) להשתמש במתודות אבסטרקטיות ולממש באופן שונה 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.

תיאור השימוש בממשק הגרפי

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

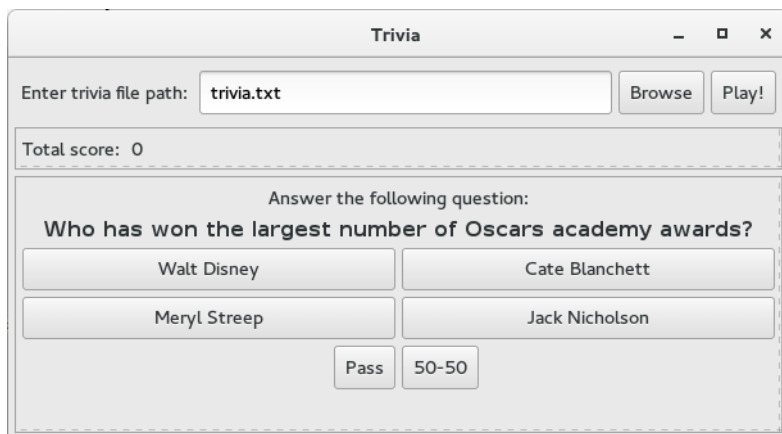


ניתן להקליד מסלול לקובץ בשדה הנתון, או ללחוץ על הכפתור Browse. במקרה הזה, ייפתח חלון לבחירת קובץ.

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.

- 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. אחרי ההודעה התוכנית תישאר פתוחה ויהיה ניתן לטעון קובץ חדש וללחוץ על play להתחלת משחק חדש (או גם play עם אותו קובץ שטענו כבר). לאחר סיום המשחק והצגת ההודעה אין צורך לממש התנהגות מיוחדת לכפתורים 50-50 passi.



תכנית לדוגמא

תכנית לדוגמא נמצאת באתר הקורס כחלק מהzip של שלד הקוד בתיקייה src. בכל מקרה של ספק לגבי אופן פעולת המשחק במצב מסויים, ניתן להיעזר בדוגמא זו שכן ההתנהגות שלה היא הקובעת. יש לחלץ את הקובץ Trivia.jar מהzip ולשמור אותו בתיקייה כלשהי במחשב. באותה תיקיה, שמרו עותק של swt.jar המתאים למערכת ההפעלה שלכם כפי שיוסבר בחלק הבא.

הורדה של SWT

ניתן להוריד את ה-jar מ-<http://www.eclipse.org/swt/> ולנוחיותכם זה הקישור משם לגרסה האחרונה של swt שניתן להשתמש בה לתרגיל - <https://download.eclipse.org/eclipse/downloads/drops4/R-4.18-202012021800/>

יש לבחור את הגרסה המתאימה של zip למערכת ההפעלה שלכם. מאחת מהאופציות הנ"ל:

Platform	Download	Size
Windows (64 bit version)	swt-4.18-win32-win32-x86_64.zip	3.9 MB
Linux (64 bit version)	swt-4.18-gtk-linux-x86_64.zip	3.6 MB
Linux (64 bit version for Power PC)	swt-4.18-gtk-linux-ppc64le.zip	3.6 MB
Linux (64 bit version for AArch64)	swt-4.18-gtk-linux-aarch64.zip	3.6 MB
Mac OSX (64 bit version)	swt-4.18-cocoa-macosx-x86_64.zip	3.4 MB

למי שיש מערכת הפעלה עם 32 ביט (אפשר לבדוק האם זה המצב לפי המדריך [הזה](#)) אפשר להוריד גרסה ישנה יותר של swt שתומכת בכך כאן: <https://archive.eclipse.org/eclipse/downloads/drops4/R-4.9-201809060745/>

Platform	Download	Size
Windows (32 bit version)	swt-4.9-win32-win32-x86.zip	4.1 MB
Windows (64 bit version)	swt-4.9-win32-win32-x86_64.zip	4.2 MB
Linux (32 bit version)	swt-4.9-gtk-linux-x86.zip	3.8 MB
Linux (64 bit version)	swt-4.9-gtk-linux-x86_64.zip	3.9 MB
Linux (64 bit version for Power PC)	swt-4.9-gtk-linux-ppc64le.zip	3.8 MB
Mac OSX (64 bit version)	swt-4.9-cocoa-macosx-x86_64.zip	3.4 MB

קובץ הזיפ שתורידו יכיל 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-Windows 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-8 java מותקנת על המחשב, יפתח חלון עם משחק הטריוויה. (אם עדיין לא עובד לכם, נסו לשנות את שם ה-jar להיות swt בלבד ללא הסיימת jar). לעיתים יש להגדיל (למתוח) את החלון הנפתח כדי לראות את כל הכפתורים. על מנת להריץ על מחשבי Mac יתכן שיהיה צורך להשתמש בפקודה:

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

מה עליכם לעשות

באתר הקורס נתונים לכם שלדים של מחלקות ה-GUI בחבילה `il.ac.tau.cs.sw1.trivia`. הוסיפו ושנו קוד לפי הצורך וראו עיניכם. יש להוסיף ל-`build path` של הפרויקט את `swt.jar` המתאים למערכת ההפעלה.

איך לטעון את קובץ `swt.jar` לפרויקט שלכם

כדי להוסיף ל-`build path` של פרויקט קובץ `jar` ב-eclipse יש ללחוץ מקש ימני על הפרויקט-<Build Path Configure `Libraries` (ניתן להגיע לשם גם ע"י מקש ימני על הפרויקט-<properties) בחלון שיפתח יש לגשת לטאב `Add External JARs..` ולבחור את `swt.jar` שהורדתם בחלק הקודם וללחוץ Apply. לאחר מכן קוד המחלקה אמור להתקמפל ויהיה ניתן להריץ את `TriviaMain`.

באתר הקורס גם נתון קובץ טריוויה תקין לדוגמא, `trivia.txt`.

פירוט

5] 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.getPathFromFileDialog` הנתונה לכם.

20] 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). לא ייתכנו שתי שאלות עם אותו טקסט ואותה קבוצת תשובות בדיוק.

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, וְאוּ במתודה `Collections.shuffle` כדי לסדר את רשימת התשובות בסדר אקראי. אתחלו את הניקוד להיות 0, ושדות אחרים לפי הצורך.

15] 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-2 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 סיימו 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 (הניקוד יכול לרדת מיידית עם הלחיצה על הכפתור או לאחר מענה על השאלה). 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. הבנה ושימוש בשלד הקוד הקיים הוא אתגר מרכזי בתרגיל.
- נדגיש שוב – בכל ספק בנוגע להתנהגות הרצויה, יש לבדוק את ההתנהגות לפי התוכנית לדוגמא ולממש את הקוד שלכם כך שיפעל באותה צורה.
- בחלק זה של התרגיל אתם רשאים לשנות את המחלקות והמתודות הנתונות לפי הצורך, מלבד פונקציית 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.
- ניתן להניח כי ה-path שמתקבל כקלט הוא תקין ומוביל לקובץ טריוויה תקין.
- יש להגיש את כל המחלקות (קבצי .java), גם אם לא שיניתם אותן. אין צורך להגיש את `swt.jar`, אבל זה לא משנה גם אם תגישו אותו.
- את חלק ה-GUI אין חובה לבדוק בהרצה בcmd ובלינוקס, מספיק שהתוכנית עובדת באקליפס (אך עדיין שימו לב לא להשתמש בimportים שלא נראו בקורס ולא להשתמש בפתרונות ל-path שספציפיים למערכת הפעלה מסוימת). על מנת כן לבדוק את עצמכם יש לארוז את הקוד שלך בקובץ `jar` בדומה לקובץ הדוגמה, ניתן להיעזר במדריך [הזה](#).

בהצלחה !