

# 10118 – תכנון מונחה עצמים בשפת JAVA

## פינוקי בית – תש"פ קיץ

מטלה 2:

הורשה, פולימורפיזם, ממשקים, קבצים, חריגות, GENRICS ו-  
COLLECTIONS

תאריך הגשה: יפורסם במשך

### מטרת התרגיל

שדרוג המערכת שכתבתם בתרגיל 1 תוך שימוש בכלים החדשים שנלמדו בשפה.

### הנחיות הגשה כלליות

1. יש לעבוד בסביבת הפיתוח Eclipse
2. הקוד חייב להיות קריא, מסודר וברור
3. הקוד חייב לעבור קומפילציה ולהיבדק. קוד שאינו מתקמפל לא יבדק!
4. קוד הפרוייקט צריך להיות בתוך פרוייקט ששמו כשם הסטודנט package ששמו כשם הסטודנט, ויש להעלות למערכת ה- moodle קובץ זיפ של הפרוייקט לאחר שבצעתם לו export.
5. ניתן להגיש את העבודה עד 7 ימים איחור אשר תגרור **הורדה של 10 נקודות**. אין צורך בבקשת אישור מיוחד להגשה באיחור בטווח זה. מעבר למועד השני לא ניתן להגיש את התרגיל משום סיבה פרט למילואים או אישור מחלה מרופא.

### יש להקפיד על הדברים הבאים

1. שמות משתנים, מחלקות ופעולות ברורים
2. שימוש בקבועים והגדרתם
3. קוד מרווח ומאוורר
4. הפרדת חלקי השאלה השונים ע"י שורת רווח: הגדרת משתנים, קלט, חישוב ולבסוף הפלט
5. הפרדה בין מחלקות הלוגיקה של המערכת (שכבת ה- Business Logic) לבין המחלקות המתעסקות בקלט ובפלט (ממשק המשתמש, שכבת ה- UI).
6. במידה ולא השתמשתם בהורשה ובפולימורפיזם בחלק הראשון, זה הזמן.
7. אבחנה בין הורשה להכלה

## עולם הבעיה

בחלק זה נמשיך את הפרוייקט מהחלק הקודם ונוסיף לו יכולות:

נוסיף אפשרות שמירה וטעינה מקבצים של כל שאלות ותשובות המבחן, כך שבכל הרצה לא יהיה צריך להזין את כל המידע מחדש. כלומר, עם היציאה מהתוכנית, ישמרו לקובץ כל שאלות המבחנים והתשובות שהוזנו במהלך ריצת התוכנית. עם תחילת התוכנית, יטענו לזכרון נתוני השאלות והתשובות.

1. תקנו את ההערות שקיבלתם למטלה 1.
2. שנו את המערכת כך שטיפול בחריגות יהיה באמצעות מנגנון ה-Exception'ים. משמע, כעת אסור שתהייה כל הדפסה למסך בתוך המחלקה, וכל האינטראקציה עם המשתמש הינה בקובץ ייעודי.
3. הגדירו ממשק לכל פעולות המערכת. התאימו את המחלקה המנהלת כך שתממש ממשק זה. המטרה: שה- UI יעבוד מול הממשק בלי קשר למחלקה שמממשת אותו בפועל.

4. נוסיף למערכת ולתפריט אפשרות של יצירת מבחן באופן ידני:
  - התוכנית תשאל את המשתמש כמה שאלות הוא רוצה שיהיו במבחן.
  - התוכנית תשאל את המשתמש איזה שאלות מתוך המאגר הוא רוצה שיהיו במבחן, ועבור כל שאלה אילו תשובות יופיעו עבורה. ניתן לבחור יותר מתשובה אחת נכונה.
  - לאחר סיום בחירת השאלות והתשובות, יש לייצר את קובץ המבחן שהוא קובץ טקסט עם השאלות והתשובות שנבחרו. בנוסף לתשובות שנבחרו, יש להוסיף תמיד את שתי התשובות הבאות: "אף תשובה אינה נכונה" ו"יותר מתשובה אחת נכונה".
  - כמו-כן יש לייצר קובץ תשובות למבחן שנוצר.
5. נוסיף לתפריט אופקציה של יצירת מבחן באופן אוטומטי:
  - התוכנית תשאל את המשתמש כמה שאלות הוא רוצה שיהיו במבחן.
  - התוכנית תבחר באופן אקראי שאלות ו- 4 תשובות עבור כל שאלה.
  - באפשרות זו לא ניתן שתהייה יותר מתשובה אחת נכונה לשאלה, אך יתכן ואף תשובה לא תהייה נכונה.
  - בסיום יש לייצר את קובץ המבחן שהוא קובץ טקסט עם השאלות והתשובות שנבחרו. בנוסף לתשובות שנבחרו, יש להוסיף תמיד את התשובה "אף תשובה אינה נכונה".
  - כמו-כן יש לייצר קובץ תשובות למבחן שנוצר.

שימו לב: שם כל קובץ מבחן שיווצר יהיה בפורמט הבא: exam\_yyyy\_mm\_yy\_hh\_mm 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.

## ובנוסף..

- המערכת הנוכחית תומכת בפתרון שמירת 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.
6. המירו את השימוש במערכים לאוספים. לבחירתכם באיזה אוספים להשתמש מצתוך אלו שלמדנו.
  7. הוסיפו אופציה של שאלה פתוחה שתכיל טקסט של השאלה וטקסט של התשובה. בעת יצירת מבחן, נשאל את המשתמש כמה שאלות פתוחות הוא רוצה במבחן. הקפידו על שימוש בהורשה ובפולימורפיזם.

## עבודה נעימה!