



# אוניברסיטת בן-גוריון בנגב

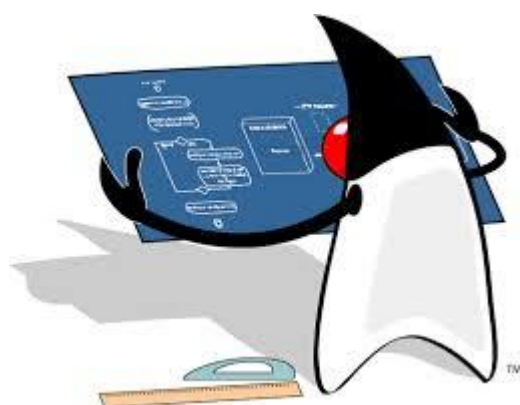
## Ben-Gurion University of the Negev

## משחק מבוך

סמסטר ב' תשע"ט

### תקציר

- חלק א' - יצירת ספריית קוד
  - מפסאודו-קוד לתכנות מונחה עצמים
- חלק ב' – שרת-לקוח
  - Streaming, קבצים, decorator
  - תכנות מקבילי עם java threads
- חלק ג'
  - Architectural Design Pattern
  - תכנות מונחה אירועים
  - GUI בטכנולוגיית JavaFX



מחבר: ד"ר אליהו חלסטצ'י  
khalastc@post.bgu.ac.il

---

## Table of Contents

---

2.....	מנהלות
3.....	הנחיות כלליות
4.....	הוראות הגשה
5.....	בדיקת הפרויקט
6.....	הקדמה
6.....	מבוך דו-מימדי
7.....	חלק א': מפסאודו-קוד לתכנות מונחה עצמים
8.....	משימה א' – אלגוריתם ליצירת מבוך
9.....	בדיקות
11.....	משימה ב' – אלגוריתם חיפוש
12.....	בדיקות
13.....	משימה ג' – Unit Testing (למידה עצמית)
13.....	משימה ד' – עבודה עם מנהל גרסאות (למידה עצמית)
14.....	דגשים להגשה
15.....	חלק ב': עבודה עם Threads-ו Streams
15.....	הקדמה
15.....	משימה א' – דחיסה של Maze ו-Decorator Design Pattern
16.....	בדיקות
18.....	משימה ב' - שרת-לקוח ו-Threads
19.....	בדיקות
21.....	משימה ג' – קובץ הגדרות (לימוד עצמי)
22.....	חלק ג': אפליקצית Desktop בארכיטקטורת MVVM, תכנות מונחה אירועים, ו-GUI
22.....	משימה א' – ארכיטקטורת MVVM
23.....	משימה ב' – תכנות מונחה אירועים ו GUI
23.....	יצירת רכיב גרפי – תזכורת
25.....	כתיבת ה-GUI
25.....	עיצוב ה-GUI
26.....	דגשים להגשה

---

## מנהלות

---

בקורס זה יינתן תרגיל חימום אחד עם חובת הגשה וללא ציון על מנת להכין אתכם לפרויקט. התרגולים במעבדה נועדו ללמד אתכם את יסודות שפת *Java*, את מה שיש לה להציע ואת דרכי העבודה המומלצות יחד עם הדגמת החומר הנלמד בהרצאה. הכלים והידע שתלמדו במעבדה נדרשים לכתיבת הפרויקט וימשו אתכם רבות בהמשך דרככם.

לפרויקט שלוש אבני דרך, מועדי הגשתם המעודכנים רשומים באתר ה-Moodle של הקורס.

**משקל כל חלק בפרויקט 33% ממשקל הפרויקט הכולל.**

**כל חלקי העבודה הינם להגשה בזוגות בלבד.**

כל חלק בפרויקט נבנה בהתבסס על החלק שקדם לו לכן חשוב לעמוד בדרישות בצורה הטובה ביותר. עמידה בדרישות בחלק מסוים בפרויקט יקל עליכם בחלקים שיבואו אחריו.

הפרויקט מאורגן בצורה מרווחת מאוד כך שיש מספיק זמן לביצוע כל מטלה. כבר ניתנו הזמנים המקסימליים עבור כל מטלה, כל בקשה להארכה קולקטיבית משמעותה לבוא על חשבון המטלות האחרות. להארכות זמן פרטניות מסיבות מוצדקות בלבד יש לפנות למתרגל הקורס.

בכלל זה נוסף שהעתקה אינה משתלמת; מעבר לסיכון להיתפס ולהישלח (המעתיק והמועתק) לוועדת משמעת, כל צורה של העתקה (גם מפרויקטים דומים קודמים) אינה שווה כלל למימוש קוד עצמאי, ובכך היא פוגעת ביכולת שלכם להפנים את החומר ותוביל לכישלון במבחן.

בהצלחה,

דודי, אביעד ודניאל.

## הנחיות כלליות

אנו ממליצים בחום לקרוא את מסמך זה מתחילתו ועד סופו ולהבין את הכיוון הכללי של הפרויקט עוד בטרם התחלתם לממש. הבנת הדרישות ומחשבה מספקת לפני תחילת המימוש תחסוך לכם זמן עקב טעויות מיותרות. "סוף מעשה במחשבה תחילה".

עבדו עם **IntelliJ** בגרסה העדכנית ביותר, כפי שמותקן במעבדות בהן מועבר בתרגול. את גרסת ה-Community החינמית, תוכלו להוריד מכאן:

<https://www.jetbrains.com/idea/download/>

עבור כל חלק בפרויקט זה, שעליו אתם מתחילים לעבוד אנו ממליצים:

1. לקרוא את כל הדרישות של החלק מתחילתם ועד סופם.
2. לדון על הדרישות עם בן/בת הזוג למטלה.
3. לחשוב איך אתם הולכים לממש את הדרישות.
4. לייצר לעצמכם **Class Diagram**.
5. לתכנן את חלוקת העבודה ביניכם.
6. להתחיל לממש.
7. לשמור ולגבות את הגרסאות השונות של הקוד שלכם במהלך העבודה במספר מוקדים שונים: **Dropbox, Email** וכו'.
8. לבצע בדיקות עבור כל קוד שמימשתם. בדקו כל מתודה על מנת להבטיח שהיא מבצעת כראוי מה שהיא אמורה לבצע. זכרו שאתם יכולים לדבג (**Debug**) את הקוד שלכם ולראות מה קורה בזמן ריצה.
9. במידה ועשיתם שינויים ו"הרסתם" משהו בפרויקט, זכרו שב-**IntelliJ** קיימת האופציה לצפות בהיסטוריה של כל מחלקה ולעקוב אחרי שינויים (מקש ימני על קוד מחלקה < Local History).
10. לשמור את התוצאה הסופית שאותה אתם הולכים להגיש במספר מוקדים שונים.

דגשים לכתיבת קוד:

- הקפדה על עקרונות ה-**SOLID** שלמדתם.
- הפונקציות צריכות להיות קצרות, עד 30 שורות ולעסוק בעניין אחד בלבד. פונקציות ארוכות ומסובכות שעוסקות בכמה עניינים הם דוגמא לתכנות גרוע.
- הפונקציות צריכות להיות גנריות (כלליות), שאינן תפורות למקרה ספציפי.
- שמות משתנים ברורים ובעלי משמעות.
- שמות שיטות ברורים ובעלי משמעות.
- מתן הרשאות מתאימות למשתנים ולמתודות (**public, protected, private**). כימוס (**Encapsulation**)<sup>1</sup>.
- שימוש נכון בתבניות העיצוב שנלמדו בכיתה, בירושה ובממשקים.
- תיעוד הקוד:
  - תיעוד מעל מחלקות, שיטות וממשקים.
  - יש לתעד שורות חשובות בתוך המימוש של השיטות.
  - הסבר על תיעוד Javadoc ניתן למצוא כאן:

[https://www.tutorialspoint.com/java/java\\_documentation.htm](https://www.tutorialspoint.com/java/java_documentation.htm)

בסיום כתיבת הפתרון:

1. הריצו את הפרויקט ובדקו אותו ע"פ הדרישות של החלק אותו מימשתם.

<sup>1</sup> <https://he.wikipedia.org/wiki/%D7%9B%D7%99%D7%9E%D7%95%D7%A1>

2. עברו שוב על דרישות המטלה ובדקו שלא פספסתם אף דרישה.
3. חשבו על מצבי קצה שאולי עלולים לגרום לאפליקציה שלכם לקרוס וטפלו בהם.

קחו בחשבון שהפרויקט שאתם מגישים נבדק על מחשב אחר מהמחשב שבו כתבתם את הקוד שלכם. לכן, אין להניח שקיים כונן מסוים (לדוגמה: D) או תיקיות אחרות בעת ביצוע קריאה וכתובה מהדיסק.

בנוסף, כאשר אתם כותבים ממשק משתמש, בין אם זה **Console** או **GUI**, קחו בחשבון שהמשתמש (או הבודק) אינו תמיד מבין מה עליו לעשות ולכן עלול לבצע מהלכים "לא הגיוניים" בנסיונו להבין את הממשק שלכם. מהלכים אלו יכולים לגרום לקריסה של הקוד אם לא עשיתם בדיקות על הקלט שהמשתמש הכניס או לא לקחתם בחשבון תרחישים מסוימים. לכן, חשוב שתעזרו למשתמש/בודק לעבוד מול הממשק שלכם ע"י הדפסה של הוראות ברורות מה עליו להקליד, על איזה כפתור ללחוץ (ומתי) ומה האופציות שעומדות בפניו. הנחיות אלו יפחיתו את הסיכויים לטעויות משתמש שעלולות לגרום לקריסה של האפליקציה ולהורדת נקודות.

**חלקי הפרויקט שאתם מגישים נבדקים אוטומטית (פרט לחלק ג' שיבדק גם ידנית) ע"י קוד בדיקה לכן חשוב ביותר להצמד להוראות ולוודא ששמות המחלקות והממשקים שהגדרתם הם בדיוק מה שהתבקשתם. הבדיקה האוטומטית אינה סלחנית לכן השתדלו להמנע מטעויות מצערות.**

לפני הגשת המטלות, בדקו את עצמכם שוב! פעמים רבות שינויים פזיזים של הרגע האחרון שנעשים ללא מחשבה מספקת, גורמים לשגיאות ולהורדת נקודות. חבל לאבד נקודות בגלל טעויות של הרגע האחרון.

---

## הוראות הגשה

---

את המטלות יש להגיש ל:

1. למערכת הבדיקה האוטומטית:

<http://subsys.ise.bgu.ac.il/submission/login.aspx>

- \* בחרו סטודנט אחד מתוך הצוות שדרך חשבונו תעלו את העבודות.
- \* למען הסדר הטוב, אתם מתבקשים להעלות את העבודה רק דרך החשבון של הסטודנט הנבחר (ולא מחשבונות שונים בכל פעם).

**\*\* המטלות יכתבו ב-Java Language Level 8 בלבד.**

הגישו את התוצרים כקובץ **Zip** המכיל:

1. תיקיית הפרויקט שלכם ב-IntelliJ במלואה! התיקיה צריכה להכיל:

- a. תיקיית **src** קוד המקור.
- b. תיקיית ה-idea.
- c. תיקיות נוספות כגון **resources**.
- d. וכו'..

2. קובץ ה-**JAR** של הפרויקט שלכם.

a. הוראות יצירה ב-IntelliJ: <https://www.youtube.com/watch?v=3Xo6zSBgdgk>

- בתיקיה שאתם מגישים מחקו קבצים ותיקיות מיותרות כגון קבצים זמניים או קבצים השייכים ל-Git. (תיקיה נסתרת), התיקיה המוגשת צריכה להיות רזה יחסית.
- במידה ואתם משתמשים בקבצי מדיה, בחרו בקבצים הרזים ביותר (מבחינת גודל קובץ) המספיקים לכם.