

תוכנה 1 – חורף 2024/2025

תרגיל מספר 5

הנחיות כלליות:

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

את התרגיל הבא צריך להגיש באופן הבא:

- הגשה במערכת ה-Git תבצע על פי ההנחיות שראיתם בתרגול 1. צרו את ה repository שלכם מתוך הקישור הבא:

<https://classroom.github.com/a/QSmY2efF>

יש לוודא שבתיקיית הגיט שלכם נמצאים הקבצים הבאים:

- a. קובץ פרטים אישיים בשם details.txt המכיל את שם המשתמש שלכם ב Moodle ואת מספר תעודת הזהות שלכם.
 - b. תיקיה בשם polynomial ובתוכה היררכיית תיקיות שמכילה את קבצי הג'אווה של Polynomial
 - c. תיקיה בשם riddles ובתוכה היררכיית תיקיות 4 התיקיות a,b,c,d, שבכל אחת מהן קבצי הג'אווה הרלוונטיים (שימו לב שאלה תיקיות שקיימות כבר בשלד שקיבלתם).
 - d. היררכיית התיקיות צריכה להיות בדיוק באותו אופן שקיבלתם אותה. כלומר, הריפוסטורי שתגישו יראה בדיוק כמו הריפוסטורי של השלד, רק שיכיל בתוך קבצי java את הקוד שלכם.
- הגשה במערכת ה Moodle (<http://moodle.tau.ac.il/>): עליכם להגיש את קובץ הטקסט assignment.txt ובו קישור ל repository האישי שלכם.
 - נא לא להשתמש בפקודה System.exit()! היא מחבלת בבדיקות אוטומטיות. אין כל צורך לעשות בה שימוש, כאשר תוכניות יכולות להסתיים ע"י הגעה לסוף מתודת ה-main.

ייבוא הפרויקט:

מומלץ ליצור 2 פרויקטים שונים. אחד עבור polynomial ואחד עבור riddles.

בריפוסטורי שקיבלתם יש 2 תיקיות מתאימות עם השמות האלו, ובכל אחת מהן תיקיית src.

עבור כל אחד מהפרויקטים קחו את תיקיית ה-src המתאימה, כאשר הנחיות יבוא הפרויקט הן אותן ההנחיות כמו בתרגילים הקודמים.

המחלקות לחלק א נמצאות תחת polynomial, המחלקות לחלק ב נמצאות תחת riddles.

חלק א' (60%) – מימוש מחלקה על פי מפרט

בחלק זה של התרגיל נממש מחלקה בשם Polynomial המייצגת פולינום עם מקדמים ממשיים. המחלקה תתמוך בשירותים הציבוריים אשר מופיעים בקובץ Polynomial.

המשימה:

ממשו את המחלקה Polynomial בצורה יעילה תוך שימוש במערכים בלבד. אין להשתמש במבנה נתונים נוסף (**התבססו על השלד**). שימו לב מותר להוסיף שדות וניתן להגדיר שירותי עזר.

חשוב: במידה והמתודה מחזירה פולינום, יש ליצור פולינום חדש ולא לשנות את הקיים. לדוגמא עבור המתודה `add`, יש לחבר את הפולינום הקיים עם הפולינום שניתן כארגומנט ולהחזיר פולינום חדש, מבלי לשנות את הפולינום הקיים או את הפולינום שניתן כארגומנט.

שימו לב כי אנו מחשיבים את הדרגה של פולינום האפס בתור אפס (כמו גם הדרגה של כל פולינום שיש לו מקדם חופשי בלבד).

ניתן לבדוק את עצמכם בעזרת פונקציית ה-`main` הנתונה במחלקה `Test`. שימו לב כי בדיקה זו הינה חלקית בלבד. ניתן לשנות ולהוסיף לה בדיקות כרצונכם (יש להגיש את הקובץ `Test`).

חלק ב' (20%) – חידות ג'אוה ותיקון מחלקות קיימות

בכל סעיף של חלק זה תקבלו חבילה ובה מספר מחלקות. חבילות אלה מצורפות לתרגיל באתר הקורס. עליכם לשנות את הקוד **בהתאם להנחיות**, כדי לקבל את התוצאה הנדרשת. יש להגיש את כל המחלקות (כולל אלה שלא שיניתם בהם דבר, וכמובן, הקוד המתוקן). אין רווחים בין השורות של ההדפסות.

1. החבילה `il.ac.tau.cs.sw1.riddle.a` מכילה שתי מחלקות, `A` ו-`B`.

`B` היא תכנית המקבלת כארגומנט מס' שלם. עליכם לשנות את הקוד בתוך `printA()` ב-`A` **בלבד** כך שבהרצת פונקציית ה-`main` ב-`B` יודפס המספר שניתן כקלט בין `A1` ל-`A2`. לדוגמא, אם הקלט הוא 15, יודפס:

B

A1

15

A2

- **מותר:** לשנות את הקוד בתוך `A.printA()`.
- **אסור:** לשנות את `B`, את חתימת `printA()`, וקוד ב-`A` שנמצא מחוץ ל-`printA()`.

2. החבילה `il.ac.tau.cs.sw1.riddle.b` מכילה שלוש מחלקות, `A`, `B` ו-`C`.

`C` היא תכנית המקבלת כארגומנטים שלוש מחרוזות. עליכם להשלים את מימוש המתודות `printA`, `printA2` ו-`printA3` ב-`A` כך שבהרצת פונקציית ה-`main` ב-`C` יודפסו 3 המחרוזות בין הכוכביות. לדוגמא, אם הקלט לתכנית הוא `hello world bye`, יודפס:

```
hello

***

world

***

bye
```

- **מותר:** לשנות את הקוד בתוך `printA`, `printA2` ו-`printA3`.
- **אסור:** לשנות את `B` ו-`C`, את חתימות המתודות `printA`, `printA2` ו-`printA3`, וקוד ב-`A` שנמצא מחוץ למתודות הנ"ל.

3. החבילה `il.ac.tau.cs.sw1.riddle.c` מכילה שתי מחלקות, `A` ו-`B`.

`B` היא תכנית. עליכם לשנות את חתימות המתודות והשדות ב-`A` כך ש-`A` (א') הקוד יתקמפל ללא שגיאות, ו-`B` (ב') תדפיס, כשורה אחרונה, `success!`. ייתכן שיודפסו שורות נוספות לפני שורה זו, המשמשות לבקרה בלבד (כל עוד מודפס `success!` הפתרון נכון).

- **מותר:** לשנות את חתימות המתודות ואת השדות, כולל: שינוי נראות (`public` ל-`private` ולהפך), הוספת והורדת `static`, שינוי טיפוס ההחזרה של מתודה, ושינוי הארגומנטים למתודה.
- **אסור:** לשנות את `B`, לשנות קוד בתוך מתודות `A`, לשנות את שמות המתודות ב-`A` ולשנות את ערך השדה `k`.

4. החבילה `il.ac.tau.cs.sw1.riddle.d` מכילה שתי מחלקות, `A` ו-`B`.

`A` היא תכנית המדפיסה מס' שלם. בתוך קוד `A` מופיעות 4 קריאות לפונקציה `setI` של מחלקה `B`, עם הארגומנטים `k`, `I`, `B`, `j`, ו-`1` בהתאמה. עליכם לשנות את הארגומנטים של `setI` כך שהקוד יתקמפל ללא שגיאות, ובסופו של דבר יודפס המס' 210. עדיין, הארגומנט של כל קריאה חייב להיות `I`, `j`, `k` או `1`.

- **מותר:** לשנות את הארגומנטים של `setI`, למשל במקום בו הופיע `setI(j)` אפשר לשנות ל-`setI(k)`. אם יש צורך ניתן להוסיף שם מחלקה או מופע לפני שדה, למשל `B.I` או `this.I`.
- **אסור:** להשתמש במשתנים/שדות מלבד הארבעה הנ"ל, או במס' טבעיים. אין לשנות את `B` או כל קוד ב-`A` מלבד הארגומנטים המועברים ל-`setI`.

חלק ג' (20%) – דיזיין

בחלק זה נתרגל עיצוב (Design) של תוכנה. את הפתרון יש להעלות כקובץ `pdf` לתיקיית ה-`git` שלכם במקום הקובץ `designSol.pdf` שנמצא בראש תיקיית הגיט שלכם לצד הקובץ `details.txt`.

בשאלה זו אין צורך לספק מימושים, אלא רק את חתימות הפונקציות, הנראות שבחרתם לכל פונקציה והאם הפונקציה היא פונקציה אבסטרקטית, פונקציה סטטית, או פונקציה דורסת. לצורך שאלה זו, לפונקציות שאינן אבסטרקטיות ניתן לספק מימוש ריק.

אם הוספתם שדות, ציינו את הנראות שלהם, וכיצד (והאם) ניתן לשנות את תוכנם (האם יש צורך בפונקציות `getter / setter`). אם בחרתם שמחלקה תממש מנשק, או שתירש ממחלקה אחרת, ציינו זאת בצורה ברורה. ניתן גם להציג את התשובה בתור דיאגרמה של מחלקות/ מנשקים.

הסבירו בצורה ברורה ועניינית את מבנה המחלקות והמנשקים שבחרתם עבור בעיה זו. מותר להוסיף מחלקות או מנשקים לבחירתכם, וכן מתודות שלא צוינו בדרישות.

עליכם לעצב את מודול ההודעות של תוכנת `chat` כלשהי (דוגמת `Telegram`, `Whatsapp` וכדומה).

- על התוכנה (כרגע) לתמוך בסוגי ההודעות הבאים: מלל (טקסט), תמונה, הודעות קוליות, סרטונים ומסמכים.
- על ה-`design` לאפשר הוספה עתידית של סוגים נוספים של הודעות.
- הניחו שה `backend` של התוכנה כבר קיים. כלומר יש מודול (Module) שמעבד ומציג את ההודעות למסך (אין צורך להגיש אותו). הוא מצפה למצוא בכל הודעה את נתוני המסגרת הבאים: שולח ההודעה, מען ההודעה וזמן השליחה.
- הניחו שקיים מנשק/מחלקה `User` וזה הטיפוס שמייצג את שולח ההודעה ומען ההודעה.
- זמן השליחה מיוצג כמחרוזת.
- כמובן שלכל עצם שמייצג הודעה יש שדה שמחזיק את המידע (התוכן) של ההודעה הספציפית. נתעלם בשאלה זו מתחזוקת התוכן וייצוגו.

- כל הודעה תכיל פונקציית draw שמציגה את ההודעה על המסך.¹ כדי להציג הודעה על המסך, אפשר ורצוי להשתמש במנשק IDrawable (שנתון לכם בשלד). הלוגיקה להצגת כל סוג של הודעה היא ייחודית לסוג ההודעה
- כל הודעה ניתנת למחיקה. הלוגיקה למחיקה זהה לכל ההודעות.
- כל הודעה תהיה ניתנת להורדה. הלוגיקה להורדת כל סוג של הודעה היא ייחודית לסוג ההודעה.
- הודעות קוליות וסרטונים ניתן לנגן. הלוגיקה לניגון כל סוג של הודעה כזאת היא ייחודית לסוג ההודעה.
- כל הודעה ניתנת להצפנה בפרוטוקול DES אבל הודעה מסוג "מסמך" נצפין בפרוטוקול AES.
- ייתכן שבהמשך נרצה לתמוך בהודעות מסוגים נוספים. הסבירו כיצד נעדכן את ה design עבור סוג חדש של הודעה, ומה נדרש לממש.

ראו את IDrawable בעמוד הבא.

```
public interface IDrawable {

    /*
     * Draws the content of the object in the temp framebuffer
     * (which will then be able to be copied into the screen framebuffer)
     */
    public void draw();

    /* Returns the width of the content (in pixels) */
    public int getContentWidth();

    /* Returns the height of the content (in pixels) */
    public int getContentHeight();
}
```

בהצלחה!

¹ הדרישות בתרגיל הן הפשטה מאוד גסה של המציאות. בפועל כתיבה של תוכנית כזו מסובכת בהרבה. כתרגיל מחשבתי, ממליצים לחשוב כיצד תוכנה כזו תמומש בפועל.