

# תוכנה 1 – חורף 2022/23

## תרגיל מספר 1

### הנחיות כלליות:

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

את התרגיל הבא צריך להגיש באופן הבא:

- הגשה במערכת ה-Git. יש להתקין Git, להרשם ל-GitHub וליצור SSH-key על המחשב האישי שלכם, על פי ההנחיות בתירגול 1. לאחר מכן, צרו את ה repository שלכם מתוך הקישור הבא:  
<https://classroom.github.com/a/9hKmoRbZ>  
יש לוודא שבתקיית 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. קבצי ה-java של התוכניות אותם התבקשתם לממש. בתרגיל הנוכחי ישנו קובץ java יחיד בשם Assignment1 נמצא בתקיית src.

- הגשה במערכת ה Moodle (<http://moodle.tau.ac.il/>): עליכם להגיש את קובץ הטקסט assignment.txt ובו קישור ל repository git האישי שלכם. הקובץ צריך להכיל שורה אחת בדיוק, ללא מלל נוסף. לדוגמה, עבור תרגיל 1 הקובץ יכול את השורה הבאה, כשבמקום `githubUser` יופיע המשתמש שלכם ב github:

`https://github.com/software1course2223a/hw1-githubUser.git`

- בדקו את עצמכם:

על שרת ה nova (הנחיות לשימוש: [https://www.cs.tau.ac.il/system/ssh\\_servers](https://www.cs.tau.ac.il/system/ssh_servers)) עליכם לבצע את הפקודות הבאות לוודא שהקוד שלכם מתקמפל. זהו שלב הכרחי טרם ההגשה. במידה שהקוד שלכם לא יתקמפל, התרגיל לא יבדק ולא תקבלו עליו ניקוד. על מנת להריץ את הקוד על שרתי האוניברסיטה, עליכם להתחבר לשרת באמצעות הפקודה הבאה: `tauUser` הוא שם המשתמש שלכם במערכת האוניברסיטאית.

```
ssh tauUser@nova.cs.tau.ac.il
```

לאחר ההתחברות, יש לייצר SSH-key על nova לפי ההנחיות ב:

<https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

שימו לב לבחור בהנחיות עבור מערכת ההפעלה **Linux**.

יש להוריד את התרגיל שלכם מה github ע"י הרצת הפקודה הבאה:

```
git clone git@github.com:software1course2223a/hw1-githubUser.git
```

כעת, עליכם להריץ מספר פקודות על מנת לאפשר לכם גישה להתקנה של java17:  
פעם אחת בלבד עליכם להריץ את הפקודה:

```
udocker-copy software1_container
```

בכל התחברות לשרת לצורך שימוש ב java, הריצו את הפקודה:

```
udocker run --bindhome software1_container
```

לאחר מכן, עליכם להיכנס לתיקיה שבה מופיע הקוד ע"י ביצוע הפקודה (הקוד נמצא בתיקיה src שנמצאת בתיקיה hw1-githubUser שאותה הורדתם מ Github).

```
cd hw1-githubUser/src/
```

על מנת לבצע בדיקה נקיה, עדיף למחוק את קבצי ה class הקודמים שאולי היו בתיקיה. אם לא נמצאו קבצי class, הפקודה rm תדפיס: rm: No Match.

```
rm *.class
```

כעת, עליכם לקמפל את קובץ התרגיל אותו אתם מגישים ע"י הפקודה javac:

```
javac *.java
```

(שימוש ב \* מקמפל את כל הקבצים עם סיומת java שנמצאים בתיקיה ממנה אתם מריצים את הקובץ).

אם הפקודה javac לא ייצרה פלט, קובץ ה java שלכם התקמפל בהצלחה.

ניתן לוודא שנוצר קובץ class ע"י שימוש בפקודה ls המציגה את תוכן התיקיה ממנה היא נקראת:

```
ls
```

הפלט של ls אמור להכיל את שני קבצים בשם Assignment1 – אחד עם סיומת java (הקובץ שהורדתם מ Github) והשני עם סיומת class – הקובץ שנוצר ע"י הקומפיילר. כעת, תוכלו להריץ את התוכנית שלכם ע"י הפקודה java. בדוגמא המצורפת, התוכנית קיבלה כקלט 3 ארגומנטים. הפלט של הריצה צריך להיות תואם למה שעליכם לממש.

```
java Assignment1 a 1
```

אחרי שהשלמתם את הבדיקה, אפשר למחוק את התיקיה (זה כמובן לא חובה)

```
cd ../../
```

```
rm -rf hw1-githubUser
```

על מנת לצאת מה"סביבה" שבה מותקנת java17, ולחזור לממשק הרגיל של העבודה בשרתי האוניברסיטה (גם זה לא חובה), הריצו:

```
exit
```

דוגמה עבור משתמש github בשלם software1forever:

- הקלטים שמוזנים ע"י המשתמש צבועים בצהוב.
- עבור חלק מהפעולות שבהן יש פלט ארוך, הפלט מובא בחלקו והחלק החסר הוחלף ברצף של נקודות.

```
(base) *****:~ *****$ ssh *****@nova.cs.tau.ac.il
*****@nova.cs.tau.ac.il's password:
Last login: Wed Oct 26 15:45:14 2022 from gate.tau.ac.il
nova:~> git clone https://github.com/software1course2223a/hw1-
software1forever.git
Cloning into 'hw1-software1forever'...
.....
Checking connectivity... done.
nova:~> udocker-copy software1_container
Creating udocker repository in *****
.....
Creating a copy of software1_container container in your repository ...
Please wait...
Now you can run the copied container like this:
  udocker run --bindhome software1_container
nova:~> udocker run --bindhome software1_container
Warning: non-existing user will be created
Warning: no command assuming: ['/bin/bash', '-i']
.....
executing: bash
```

```
root@nova:~# cd hwl-softwareforever/src
root@nova:~/hwl-softwareforever/src# rm *.class
rm: cannot remove '*.class': No such file or directory
root@nova:~/hwl-softwareforever/src# javac *.java
root@nova:~/hwl-softwareforever/src# ls
Assignment1.class  Assignment1.java
root@nova:~/hwl-softwareforever/src# java Assignment1 a 1
New char is b.
root@nova:~/hwl-softwareforever/src# java Assignment1 b 3
New char is e.
root@nova:~/hwl-softwareforever/src# cd ../../
root@nova:~# rm -rf hwl-softwareforever/
root@nova:~# exit
nova:~>
```

**שימו לב:** חשוב מאד להקפיד על פורמט ההגשה. כלומר, יש לקרוא לקבצים בדיוק לפי ההנחיות שקיבלתם. כמו כן אין לצרף תיקיות או קבצים נוספים. **אי עמידה בהנחיות ההגשה תגרור הורדה משמעותית בניהוד!**

נדגיש שוב: בתיקיית הגיט שלכם יופיעו קובץ `details.txt`, ותיקיית `src` שבתוכה קובץ `Assignment1.java`. בנוסף, יופיעו קבצי מערכת כגון `.gitignore` ו-`README`.

---

## 1. הכרות עם Eclipse

- ✓ התקינו Java ואת סביבת העבודה של Eclipse על פי ההוראות בקישור הבא:  
<https://courses.cs.tau.ac.il/software1/2223a/misc/workenv.pdf>
- ✓ עיינו בסעיפים 9-5 ו-15 של מדריך האקליפס שבקישור הבא:  
<http://www.vogella.com/tutorials/Eclipse/article.html>

## 2. תוכנית לדוגמא

ממשו את התוכנית Assignment1. התוכנית מקבלת תו יחיד (character) ומספר שלם (integer). את התו נכנה chr ואת המספר השלם נכנה offset.

התו chr יכיל אות קטנה באנגלית (בין a ל z), והתוכנית תדפיס את האות שנמצאת במרחק offset מ chr. הניחו כי הקלט חוקי, כך שהתו במרחק offset מ chr יהיה אות קטנה בשפה האנגלית.

אין צורך לטפל בשגיאות ובמקרי קצה, התוכנית שעליכם לכתוב היא מאוד קצרה, ונועדה להכיר לכם את תהליך הפיתוח והגשת התרגילים.

דוגמאות לקלטים ופלטים.

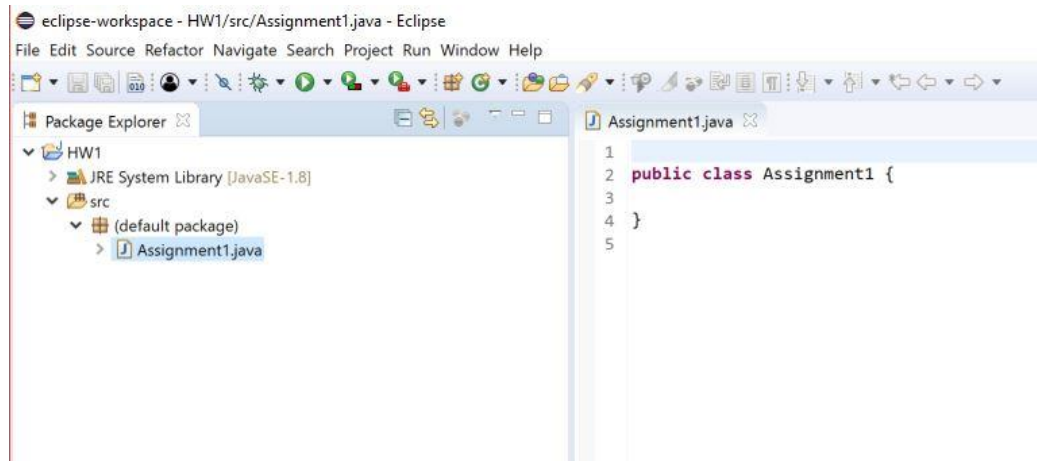
קלט לתוכנית	פלט התוכנית
a 1	New char is b.
b -1	New char is a.
p 5	New char is u.
z 0	New char is z.
z -6	New char is t.

שימו לב: בדיקת התרגילים היא **אוטומטית**, ולכן עליכם להקפיד על פורמט הפלט הנדרש!

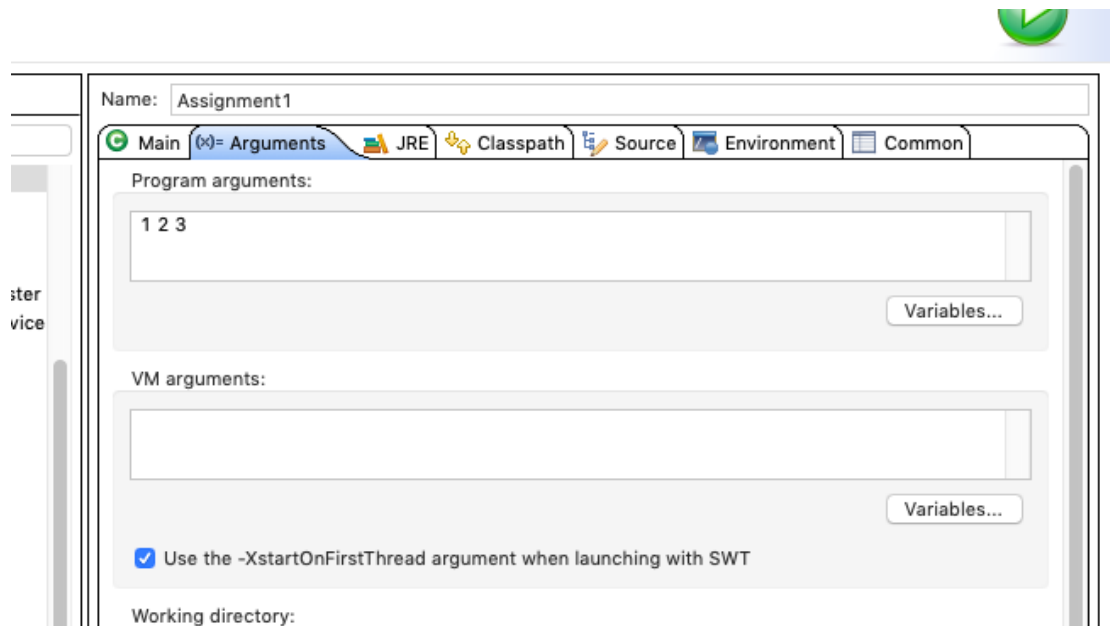
### הדרכה:

- בפונקציית ה-main של המחלקה, השתמשו במערך המחרוזות args על מנת לקבל את הארגומנטים שנשלחו בשורת הפקודה (המערך args הינו הפרמטר היחיד של פונקציית ה-main).
- ניתן להניח שהתוכנית מקבלת 2 ארגומנטים, ושניהם ניתנים להמרה ל char ול int.
- בשלד הקוד שניתן לכם מופיעה שורת קוד בהערה. עליכם להוציא שורה זו מהערה ולעשות בה שימוש בתוכנית שלכם.

לאחר קישור הפרויקט והמחלקה האקליפס נראה כך:



תזכורת: למטה מופיע צילום מסך של העברת ארגומנטים לתוכנית ב Eclipse, בדומה למה שראיתם בכיתה.



**בהצלחה !**