

תכנות מונחה עצמים – שיעור 3

בעז בן משה, benmo@g.ariel.ac.il

השלמות ב Java: Junit, Serializable, Iterator,

נושאים כלליים: Git & Github, JSON (GSON)

מתחילים Python: למה, איך, משתנים, לולאות פונק', קבצים

מבני נתונים, סביבת פיתוח, מטלה 1



Java: בדקו את עצמכם

- עקרונות OOP: אבסטרקציה, איכות, פשוטות
- בדיקות, מדריך מלא
- Serializable
- מבני נתונים <-> אלגוריתם: Iterator

מטלת כיתה:

- המחלקה Polynom (שיעור 2).



עקרונות OOP

אבסטרקציה: ייצוג העולם ע"י מחלקות

- הכמסה: encapsulation: public / private: מינימום שיטות ציבוריות
- מחזור קוד: (קצת מסובך): ירושה,
- פולימורפיזם: (רב צורתיות) לאפשר מכנה משותף לוגי.

• KIS: Keep It Simple

• השקעה במידול הבעיה – ולא (רק) בפתרונות

• אבסטרקציה

• איכות קוד: בדיקות, אחריות



Junit

כל מחלקה נבדקת:

- כל שיטה ציבורית נבדקת
- @BeforeEach, @BeforeAll, AfterEach
- אפשר לבדוק זמן ריצה (קצת מבלבל)
- יש הרבה אספקטים שאנחנו לא מבינים ב Junit – כרגע הכי חשוב זה לכתוב לכל מחלקה את הבדיקות שלה, ולתחזק אותן.



JSON

פורמט טקסטואלי לייצוג מידע (אובייקטים)

- קישורים: 1, 2, 3
- כדאי להשתמש ב GSON,
- דוגמאת הרצה: שיעור 3
- כעת ניתן לשמור למעשה כל אובייקט כקובץ בפורמט JSON (בעזרת הספרייה GSON).



GIT & Github

כלי מתקדם לבקרת תצורה

- git: קישורים: 1, 2, 3

- Github: קישורים: 1, 2, 3

- קורס קצר בנושא:

- כעת נסו להעביר את את מטלה 0 שלכם לGithub



דיון במטלה 0

מידול מדול מערכת שמתזמנת מעליות חכמות בבניין:
מבנה נתונים + אלג'

- הבנת המודל
- הבנת הממשקים
- כתיבת Testers
- מקרי קצה – לפי מיטב ההבנה שלכם (יש "נכון"
ו"לא נכון")
- לא לשכוח לתעד + כתיבת Readme



דיון במטלה 1

ממשכים את מטלה 0:

- python
- הפעם נתמקד באלג' offline
- קלט: קובץ, פלט קובץ – ניתן לבדוק
- מפתחים + מגישים ב Github.
- מומלץ להתחיל לתכנן, לחשוב על אלג' offline, להבין מה צריך ב python כדי לממש את מטלה 1





מתחילים ללמוד Python

ממשכים את מטלה 0:

- כלי פיתוח מרכזי: PyCharm, אבל גם Colab
- [מצגת](#), מחברת [1,2](#)
- [קורס מהיר](#) ב 4 שעות:
- [מדריך מלא](#)

מה צריך: משתנים, לולאות, פונקציות, קבצים, מבני נתונים, שיטות פיתוח ← מחלקות, ממשקים

