

תרגיל בית 1 – OOP וקבצים

אוף הגשה:

- במידה ובתרגיל מופיעה דוגמא של פلت מסויים, הקפידו שהפלט שלכם שלכם יהיה זהה.
- אם בתרגיל הוזכרו במפורש שמות מחלקות/משתנים וכד', או קבצים בהם יש להשתמש – יש לבדוק בשמות, ולהשתמש בקבצים שהוגדו. יש למש את השאלות ב `java`
- יש לרשום את שם ות.ז. המציגים בראש כל קובץ המהווה חלק מפתרון התרגיל. (לדוגמא: אם בניתם 3 אד השם ות.ז. יופיע בראש שלוש המחלקות).
- במידה והمطلوب מכילה יותר מקובץ אחד, יש לכוז את כל הקבצים לקובץ zip. (ולא פורמטים אחרים כמו `jar`), כאשר שם הקובץ המכווץ כולל את ת.ז. המציגים. למשל, אם הגישו את העבודה בעלי ת"ז 11111 ו- 22222, שם הקובץ יהיה `zip.zip`.
- את הקוד יש לפתח ב-**GIT** ולצרף **קישור ל-GIT** להגשה בלימוד.
- ההגשה בזוגות בלבד (אלא אם כן אושר אחרת ע"י המרצה)
- יש להגיש את הפתרון עד למועד שנקבע בלימוד
- חלק מהנבדק בתרגיל הוא עמידה בנקודות הנ"ל. הקפידו לעמוד בהן.

בהצלחה!

בעולם של משחקי מחשב וدمיוון, לכל שחקן יש רמה, לכל חפץ יש ערך, ולכל עולם – שחקנים שממלאים אותו חיים. בעבודה זו נבנה מערכת של מחלקות ב-Java שמתארת עולם פנטזיה הכלול שחקנים (Player), חפצים (Item) ועולמות משחק (GameWorld).

תרגיל 1 – מחלקה Item (חפץ)

שדות: (private)

- שם החפץ — String name
- ערך החפץ (בנקודות או בסוף) — int value
- האם החפץ קסום — boolean isMagical

בנייה:

- ברירת מחדל (name = "Unknown", value = 0, isMagical = false).
- עם פרמטרים.

מתודות:

- מחזירה תיאור של החפץ. — toString()

דוגמה: Name: Sword

Value: 3

Magical item

- מחזירה את הערך. — getValue()

•מעדכנת את השדה value. אם המתודה מקבלת ערך שלילי, היא מאתחלת את value ל- 0.

- מחזירה האם החפץ קסום. — isMagical()

- מחזירה את שם החפץ. — getName()

תרגיל 2 – מחלקה-על : Character

יצור מחלקה-על כללית בשם Character, שתכיל תכונות משותפות לשחקנים מסוגים שונים.

מחלקה Character

שדות: (private)

- שם הדמות. — String name
- מד בריאות. (מספר מ 0 עד 10). — Int health
- רמה. (מספר מ 0 עד 5). — Int level

בנאים:

- ברירת מחדל (name = "Unknown", health = 0, level = 0).
- עם פרמטרים. בדקו את level + health.

מתודות:

- getName()
- getHealth()
- getLevel()
- levelUp() — מעלה את הרמה באחד. בדקו הגבול של רמה.
- heal(int amount) — מעלה את ערך הבריאות של השחקן בהתאם לערך שהתקבל.
- toString() — תיאור כללי של הדמות.

דוגמה: Name: Elara the Enchanter

Health: 10

Level: 8

תרגיל 3 – מחלקה Player

היא מחלקה ירושת מחלקה Character ומוסיפה אליה את יכולת האחסון של חפצים.

שדות: (private)

- רשימת חפצים. — Item[] inventory
- את השאר הוא ירוש מחלקה Character.

בנאים:

- בניין עם פרמטרים שמקבל רק שם, בריאות, רמה ואתחל את inventory למערך רק.
- בניין עם פרמטרים שמקבל את כל השדות של המחלקה (Item[] inventory).
- שני הבנאים משתמשים בבניין של המחלקה Character באמצעות קראיה ל- super.

מתודות:

- addItem(Item item) — מוסיף חפץ למערך.
 - calculateInventoryValue() — מחזירה את סך הערךם של כל החפצים.
 - hasMagicalItem() — מחזירה true אם יש לפחות חפץ מסוים.
- () — מציגה את כל המידע על השחקן, כולל שמות החפצים. המתודה משתמשת במתודת `toString` של המחלקה העיל(`super.toString()`) ומוסיפה עליה מידע נוסף בהתאם לשדות החדש של המחלקה. דוגמה:
- Name: Elara the Enchanter
Health: 10
Level: 2
Inventory: Grimoire | Ice Sword

תרגיל 4 – מחלקה **Enemy**

יצירת מחלקה חדשה שמייצגת אויב במשחק. היא גם מחלקה יורשת ממחלקה Character ומוסיפה עליה רמת נזק, והאם הוא בוט.

שדות:

- int damage — כמה נזק הוא עשו. (מספר מ 1 עד 5).

בנייה:

- בניין עם פרמטרים שמקבל את כל השדות הדרושים, ומשתמש בבניי של המחלקה super Character באמצעות קריאה ל-.

מתודות:

- getDamage()
- () — כולל שם, בריאות, רמה ונזק.

DOGMA: Name: Goblin

Health: 10

Level: 4

Damage: 3

תרגיל 5 – מחלקה GameWorld

שודות:

- שם העולם — String name.
- כל השחקנים — Player[] players.
- כל האויבים — Enemy[] enemies.
- רמת קושי(int difficulty) — (Easy : 1, Medium: 2, Hard: 3).

בנייה:

- בירית חדש (שם "Crystalpath" רמה 1, מערך ריק לשחקנים ואויבים)
- עם פרמטרים.

מתודות:

- addPlayer(Player player) — מוסיפה שחקן לעולם.
- printAllPlayers() — מדפיסה את כל השחקנים בעולם.

דוגמה:

----- Player 1 -----

Name: Elara the Enchanter

Health: 5

Level: 0

Inventory: Grimoire | Ice Sword

----- Player 2 -----

Name: Kael the Windcaller

Health: 5

Level: 0

Inventory: Guardian Shield | Wand of Winds

- getTotalValueOfWorld() — מחשבת את סך הערכיהם של כל החפצים אצל כל השחקנים.
- toString() — תיאור של העולם, כולל רמת קושי וכמות שחקנים.

דוגמה:

World Name: Crystalpath

Difficulty: 2

Number of players: 2

מחלקה Main (Game)

1. צור חפצים (Items) :

- צורך לפחות 2–3 חפצים עם שמות וערכים שונים.

2. צור שחקנים (Player) :

- צורך לפחות שני שחקנים, כל אחד עם שם, בריאות 10 ורמת התחליה.

הוסף להם חפצים למלאי באמצעות המתודה `addItem`.

3. צור אויב (Enemy)

4. צור עולם משחק (GameWorld) :

- אתחל עולם משחק חדש עם שם, מערכ שחקנים, אויבים ורמה.

5. הצג את כל השחקנים בעולם המשחק. `printAllPlayers`:

6. הצג ערך כולל של החפצים לכל שחקן:

- עבור כל שחקן, הדפס את שמו ואת ערך החפצים שברשותו (באמצעות `calculateInventoryValue`)

7. הרץ סימולציה קרב בין שחקן לאויב:

- כתב פונקציה בשם `simulateBattle(Player, Enemy)`.

- השוואה בין הרמות – מי שמנצח, מקבל עלייה ברמה (`levelUp`)

8. הדפס את תוצאות הקרב.

דוגמה: Elara the Enchanter - level : 4 VS Goblin - level : 2

Elara the Enchanter wins!

חלק 2 - קבצים

כתבו תוכנית הקולעת שמות של 2 קבצים:

- הקובץ הראשון (למשל CS.txt) מכיל שם פרטי, שם משפחה ות.ז. של הסטודנטים במחלקה כלשהי.
לדוגמה:

Moshe Cohen 1234

Dina Levi 1345

Yosi Svisa 2313

השורות בקובץ ממוקנות לפי ת.ז. של הסטודנט.

- הקובץ השני (למשל hedva.txt) מכיל ת.ז. ושני ציונים של הסטודנטים רשומים לקורס (למשל לחיזוא).
לדוגמה . לדוגמה:

1234 85 75

1255 56 100

1345 90 86

השורות בקובץ ממוקנות לפי ת.ז. של הסטודנט.

על התוכנית לקרוא את שני הקבצים הנ"ל, להכין דוח ולבתור אותו לקובץ zusätzlich נוסף txt.report. הקובץ יוכל ת.ז., שם פרטי, שם משפחה וציון ממוצע של כל הסטודנטים מהמחלקה רשומים לקורס. למשל, עבור שני הקבצים מלמעלה נקבל דוח הבא:

1234 Moshe Cohen 80

1345 Dina Levi 88

הערות:

- יש לעבור על כל אחד מהקבצים רק פעם אחת.
- אסור להעתיק את כל התוכן של הקובץ לבניה נתונים כלשהו (למשל לתוך מערך). מותר לשומר במשתנים רק שורות בודדות מכל קובץ.
- שימוש לב שבסACHEKA יכולם להיות סטודנטים שלא רשומים לקורס ובקורס יכולים ללמידה סטודנטים גם מחלקות אחרות.