

## Java - שאלות תרגול #4

### מחלקות (Classes)

1. מהי מחלקה? למה כדאי להשתמש במחלקות בתוכנית שלנו?

2. מהו מופע (Instance)?

3. כיצד מייצרים מופע של מחלקה? הראה דוגמא.

4. צור מחלקה בשם Dog עם השדות (Properties) הבאים:

(1) שם (string)

(2) גיל (int)

(3) גזע הכלב (string)



כעת במחלקת Main צור פונקציה בשם "createDog" ובתוכה ייצר מופע של Dog.

5. הוסף למחלקה שכתבת בשאלה 4 פונקציה בשם "bark" שכאשר קוראים לה, הכלב נובח. (הדפיסו "woof, woof!" ואת שמו של הכלב)

6. מה תפקידה של הפונקציה toString?

7. הוסף למחלקה שכתבת בשאלה 4 פונקציית toString שמדפיסה את כל השדות. במחלקת Main צור פונקציה בשם "printDog" ובתוכה צור מופע של מחלקת Dog והדפס אותו כדי לוודא שה-toString עובד כראוי.

8. צור מחלקה בשם FootballTeam (קבוצת כדורגל) עם השדות הבאים:

(1) שם (string)

(2) מדינה (string)

(3) רשימת שחקנים (מערך של string בגודל 11)

(4) דירוג קבוצה (int)



- הוסף למחלקה פונקציה בשם "changeRating" שמקבלת מספר (int) ומשנה את דירוג הקבוצה.

- הוסף למחלקה פונקציה בשם "replacePlayer" שמקבלת שני שחקנים (שחקן נוכחי ושחקן חדש) ומחליפה במערך (של רשימת השחקנים) את השחקן הנוכחי בשחקן החדש.

9. צור מחלקה בשם Meal (ארוחה) עם השדות הבאים:

(1) שם (string)

(2) טבעוני (כן/לא) (bool)

(4) מחיר (float)



- הוסף למחלקה בנאי (Constructor) שמקבל כפרמטרים את כל שדות המחלקה.

- כעת במחלקת Main צור פונקציה בשם "createMeals" ובתוכה ייצר שני מופעים של Meal באמצעות הבנאי.

10. במחלקה שכתבת בשאלה 9, סמן את הבנאי בהערה (Comment) כדי לבטל אותו. כעת השתמש ב-lombok כדי להוסיף מחדש בנאי, ופונקציית toString למחלקה.

11. מהו שדה מסוג final ? מהו שדה מסוג static ?

12. צור מחלקה בשם House (בית) עם השדות הבאים:



(1) כתובת (string) (final)

(2) שטח במטרים (float) (final)

(3) שם הבעלים (string)

- הוסף למחלקה שני בנאים: אחד שמקבל את כל השדות, ואחד שמקבל רק את כתובת ושטח הבית.

- במחלקת Main צור פונקציה בשם "createHouses" ובתוכה ייצר שני מופעים של House, כל אחד

באמצעות

בנאי אחר.

13. צור מחלקה בשם Weather (מזג האוויר) עם השדות הבאים (קבע בעצמך את סוגי המשתנים):



(1) טמפרטורה (במעלות צלזיוס)

(2) אחוזי לחות (בין 0 ל-100)

(2) מעונן (כן/לא)

(4) סיכוי לגשם (בין 0 ל-100)

- הוסף בנאי ו-toString למחלקה באמצעות lombok.

- הוסף פונקציה בשם "changeChanceOfRain" שמקבלת כפרמטר את הסיכוי לגשם ומשנה את

השדה במחלקה. שים לב - אין לאפשר מספר קטן מ-0 או גדול מ-100.

- הוסף פונקציה בשם "printTempFahrenheit" שמדפיסה את הטמפרטורה במעלות פרנהייט.

(רמז: כדי להמיר טמפרטורה ממעלות צלזיוס © למעלות פרנהייט (F) - יש להכפיל ב-1.8 ולהוסיף 32)

14. צור מחלקה בשם Movie (סרט) עם השדות הבאים:

(1) שם הסרט (final)

(2) תאריך יציאה (final) [התאריך יקבע ברגע שנוצר מופע של המחלקה]

(3) ציון (float) [מ-1 עד 10]

(4) ז'אנר (string)

(4) movieCounter (static int) [שדה שבו יספרו כמות הסרטים שנוצרו]

(5) ratingForOscar (static final int) [הציון הנדרש כדי לזכות בפרס אוסקר, הצב כאן 9]

- הוסף בנאי ו-toString למחלקה.

- הוסף פונקציה בשם resetCounter שמאפסת את ה-movieCounter.

- הוסף פונקציה בשם isWonOscar שבודקת האם הציון של הסרט עבר את הרף (ratingForOscar)

כדי לזכות באוסקר ומחזירה true או false.

## 15. \*אתגר (מחלקה שכולה שדות סטטיים ופונקציות סטטיות)

צור מחלקה בשם Memory עם השדות הבאים:

(1) מספר שלם (int)

(2) מספר עשרוני (float)

(3) מחרוזת (string)

שים לב: כל השדות צריכים להיות סטטיים (static).

- הוסף פונקציה סטטית בשם "add" שמקבלת מספר שלם (int) ומוסיפה אותו לשדה ה-int במחלקה.

- הוסף פונקציה סטטית באותו שם, שמקבלת מספר עשרוני (float) ומוסיפה אותו לשדה ה-float במחלקה.

- הוסף פונקציה סטטית באותו שם, שמקבלת מחרוזת (string) ומוסיפה אותה למחרוזת שבמחלקה.

- כעת במחלקת Main צור פונקציה בשם "addToMemory" וקרא לכל הפונקציות שיצרת במחלקת Memory (מבלי לייצר מופע).

## 16. \*אתגר

צור מחלקה בשם BankAccount (חשבון בנק) עם השדות הבאים:

(1) ownerName (שם הבעלים)

(2) balance (יתרה)

(3) creditScore (דירוג אשראי) (מ-1 עד 100)

(4) isFrozen (האם מוקפא)

- הוסף בנאי למחלקה.

- הוסף פונקציה בשם "withdraw" (משיכה) שמקבלת סכום כסף ומורידה אותו מן ה-balance.

- הוסף פונקציה בשם "deposit" (הפקדה) שמקבלת סכום כסף ומוסיפה אותו ל-balance.

- הוסף פונקציה בשם "checkBalance" שבודקת את ה-balance:

אם ה-balance נמוך ממינוס 10,000 היא תוריד 10 נקודות מן ה-creditScore.

אם ה-balance נמוך ממינוס 50,000 היא משנה את הערך של isFrozen ל-true.

- שים לב: אם החשבון מוקפא (isFrozen == true) אין לאפשר ללקוח לבצע משיכה!

## 17. \*אתגר

צור במחלקת Main פונקציה בשם "myBanks" ובתוכה צור מערך של BankAccount בגודל 3. הכנס

למערך יהיו שלושה חשבונות בנק שונים.

- עבור כל חשבון במערך, קרא לפונקציית checkBalance.

- מצא את חשבון הבנק עם דירוג האשראי הגבוה ביותר - ותן לו מענק של 1,000 ₪ (באמצעות הפונקציה desposit).

