**תרגיל 0**

**הכרת שפת Java**

1. הגדר מחלקה אבסטרקטית בשם Employee לייצוג עובד.

* שדות המחלקה יהיו:
  + שם הפרטי של העובד (firstName),
  + שם המשפחה של העובד (lastName)
  + ומספר ת.ז. שלו (id).
* הוסף למחלקה שני בנאים:

א. בנאי המקבל ערכים עבור כל שלושת השדות

ב. בנאי בירית מחבל (ז"א בנאי ללא פרמטרים) המאתחל את השם הפרטי ל: "plony", ושם המפשחה ל: "almony" ות.ז. ל-0

* הוסף למחלקה את הפעולות set/get עבור כל שדה במחלקה
* הוסף למחלקה את הפעולה toString המעמיסה (@Override) את הפעולה המוגדרת במחלקה Object, הפעולה מחזירה מחרוזת עם ערכי כל שדות המחלקה
* הוסף למחלקה את הפעולה equals המעמיסה את הפעולה המוגדרת במחלקה Object, הפעולה משווה את האובייקט שעליו היא מופעלת לאובייקט שהתקבל כפרמטר (על כל שדותיו)
* הוסף למחלקה פעולה אבסטרקטית בשםearnings המחזירה את המשכורת של העובד (בנתיים אין לנו מימוש של הפעולה)

1. הגדר מחלקה בשם HourlyEmployee שיורשת ממחלקה Employee לייצוג עובד המקבל משכורת לפי שעות.

* בנוסף לשדות הקיימים במחלקת האב, נוסיף עוד 2 שדות:
  + מספר שעות שהעובד עבד בשבוע (hours) מטיפוס int
  + התשלום לשעה (wage) מטיפוס float
* הוסף למחלקה את הבנאים הבאים:

א. בנאי המקבל ערכים עבור כל חמשת השדות

ב. בנאי ברירת מחדל המאתחל את השם הפרטי ל: “plony”, שם המפשחה ל: "almony" , ת.ז. ל-0, מספר שעות ל-0 ומשכורת לשעה ל-0

**נ.ב.** כמובן אתחול השדות שהתקבלו בירושה יהיה דרך הפעלת בנאי האב המתאים.

* הוסף למחלקה את הפעולות set/get עבור כל שדה במחלקה.
* הוסף למחלקה את הפעולה toString כנ"ל, הפעולה מחזירה מחרוזת עם ערכי כל שדות המחלקה
* הוסף למחלקה את הפעולה equals כנ"ל, הפעולה משווה את האובייקט שעליו היא מופעלת לאובייקט שהתקבל כפרמטר (על כל שדותיו)
* הוסף למחלקה פעולה earning המממשת את הפעולה המוגדרת ב-Employee ומחזירה את המשכורת של העובד (מספר השעות כפול משכורת לשעה)
* שימו לב: בבנאים ופעולות set – במידה שהמשתמש מנסה להכניס ערכים שגויים צריכים לזרוק חריגה (exception) מתאימה
* שימו לב: בפעולות מעמיסות עם צריך לעשות משהו שכבר נעשה בפונקציה המועמסת במחלקת האב – יש להפעילה בעזרת super ולא להעתיק את הקוד משם

1. הגדר מחלקה CommissionEmployee לייצוג עובד המקבל משכורת לפי הרווחים שהוא הביא לחברה. המחלקה CommissionEmployee יורשת מהמחלקה Employee.

* שדות במחלקה יהיו (בנוסף לשדות המוגרים במחלקת האב האבא):
  + רווחים ממכירות (grossSales) של שבוע מטיפוס float
  + עמלה באחוזים שהעובד מקבל מהמכירות שלו (commision) מטיפוס int
* הוסף למחלקה את הבנאים הבאים:

א. בנאי המקבל ערכים עבור כל חמשת השדות

ב. בנאי ב רירת מחדל המאתחל את השם הפרטי ל: “plony”, שם המפשחה ל: "almony" , ת.ז. ל-0, הרווחים ל-0 ועמלה ל-0

**נ.ב.** כמובן אתחול השדות שהתקבלו בירושה יהיה דרך הפעלת בנאי האב המתאים.

* הוסף למחלקה את הפעולות set/get עבור כל שדה במחלקה
* הוסף למחלקה את הפעולה toString כנ"ל, הפעולה מחזירה מחרוזת עם ערכי כל שדות המחלקה
* הוסף למחלקה את הפעולה equals כנ"ל, הפעולה משווה את האובייקט שעליו היא מופעלת לאובייקט שהתקבל כפרמטר (על כל שדותיו)
* הוסף למחלקה פעולה earning המממשת את הפעולה המוגדרת ב-Employee ומחזירה את המשכורת של העובד (העמלה חלקי 100 כפול מכירות)
* שימו לב: בבנאים ופעולות set – במידה שהמשתמש מנסה להכניס ערכים שגויים צריכים לזרוק חריגה (exception) מתאימה
* שימו לב: בפעולות מעמיסות עם צריך לעשות משהו שכבר נעשה בפונקציה המועמסת במחלקת האב – יש להפעילה בעזרת super ולא להעתיק את הקוד משם

1. הגדר מחלקה BasePlusCommissionEmployee לייצוג עובד המקבל משכורת קבועה בתוספת אחוז מהרווחים שהוא הרויח. המחלקה BasePlusCommissionEmployee יורשת מהמחלקה CommissionEmployee .

* שדות במחלקה יהיו (יהיו (בנוסף לשדות ששהיא יורשת):
  + משכורת הבסיס של העובד (baseSalary) לשבוע מטיפוס float
* הוסף למחלקה את הבנאים הבאים:
* א. בנאי המקבל ערכים עבור כל חמשת השדות
* ב. בנאי ב רירת מחדל המאתחל את השם הפרטי ל: “plony”, שם המפשחה ל: "almony" , ת.ז. ל-0, הרווחים ל-0 ועמלה ל-0 ומשכורת הבסיס ל-0

**נ.ב.** כמובן אתחול השדות שהתקבלו בירושה יהיה דרך הפעלת בנאי האב המתאים.

* הוסף למחלקה את הפעולות set/get עבור כל שדה במחלקה
* הוסף למחלקה את הפעולה toString כנ"ל, הפעולה מחזירה מחרוזת עם ערכי כל שדות המחלקה
* הוסף למחלקה את הפעולה equals כנ"ל, הפעולה משווה את האובייקט שעליו היא מופעלת לאובייקט שהתקבל כפרמטר (על כל שדותיו)
* הוסף למחלקה פעולה earning המממשת את הפעולה המוגדרת ב-Employee ומחזירה את המשכורת של העובד (משכורת הבסיס + העמלה חלקי 100 כפול מכירות)
* שימו לב: בבנאים ופעולות set – במידה שהמשתמש מנסה להכניס ערכים שגויים צריכים לזרוק חריגה (exception) מתאימה
* שימו לב: בפעולות מעמיסות עם צריך לעשות משהו שכבר נעשה בפונקציה המועמסת במחלקת האב – יש להפעילה בעזרת super ולא להעתיק את הקוד משם

1. הגדר מחלקה Payroll הבודקת את תקינות כל המחקלות המוגדרות לעיל . המחלקה מכילה פעולה סטטית main המבצעת את הפעולות הבאות:

* הגדרת מערך שיכול להכיל 3 עובדים שונים
* הכנסה למערך של עובדים:
  + באינדקס 0 יופיע עובד של HourlyEmployee
  + באינדקס 1 יופיע עובד של CommissionEmployee
  + באינדקס 2 יופיע עובד של BasePlusComissionEmployee
* לולאה העוברת על כל העובדים השמורים במערך ומדפיסה את הפרטים של כל עובד כולל המשכורת שלו לשבוע. במידה שזה עובד מהסוג שמקבל גם משכורת בסיס וגם עמלה על מכירותיו (BasePlusCommissionEmployee) תוסיף לעובד בונוס של 10% למשכורת שלו
* העובדים הנ"ל יקבלו נתונים לא אפסיים לשדותיהם והפלט צריך להציג עבודה תקינה של התוכנית, יש לדאוג שהפלט של המספרים הלא שלמים יהיה בפורמט מתאים (זאת אומרת – שתי ספרות אחרי נקודה עשרונית במשכורת)

**נ.ב.** לגבי חריגות: בשום מקרה אסור לזרוק חריגה מסוג Exception אלא חייבים למצוא חריגה מתאימה מבין החריגות המוגדרות בספריות Java. שמות כל סוגי החריגות המתאימות לכם בספריות Java מכילות סיומת "Exception".