

עבודת בית 4

תכנות מונחה עצמים

מחלקות מיוחדות

חבילות

חריגות ושגיאות

קבצים

אנומרציות (enums)

רשומות (records)

חבילות (packages)

חריגות ושגיאות

נתיבים, קבצים וספריות

המחלקה למדעי המחשב

המכללה האקדמית להנדסה ע"ש סמי שמעון

הנחיות

מועד פרסום: 04.08.2024 מועד אחרון להגשה: 20.08.2024

מתרגלת אחראית: חן קהלון

מועד הגשה

1. ההגשה היא עד סוף יום ההגשה, כלומר עד השעה 23:59 ביום המצוין לעיל כמועד אחרון להגשה. אל תחכו לרגע האחרון. תכננו את זמנכם בהתאם. הגישו לפני.
2. איחור במועד ההגשה יגרור הורדה של ציון, 5 נק' לכל יום איחור או חלק ממנו. בכל מקרה לא יהיה ניתן להגיש מעבר ל-2 ימי איחור ממועד ההגשה דלעיל.

אופן הגשה

3. קראו היטב את השאלות. עליכם לענות על כל השאלות בעבודה זו.
4. הגשת העבודה תהיה דרך אתר הקורס במודל בלבד. הגשת העבודה היא ביחידים.
5. כיצד להגיש?
- a. כל הקבצים המוגשים יהיו קבצי java, כלומר עם סיומת java. כל מחלקה צריכה לשבת בקובץ משלה עם אותו שם כמו שם המחלקה. כל הקבצים הקשורים למשימה 1 ישבו בספריה (package set) ומשימה 2 ישבו בספריה (package menu).
- b. יש להגיש רק תוכנית שעוברות קומפילציה תקינה ללא שגיאות על קומפיילר שפת java.
- c. את התשובות לשאלות הפתוחות יש להגיש בקובץ PDF בשם ex4answers.
- d. כיצד להעלות הקבצים למודל. למודל יש להעלות קובץ אחד:
יש ליצור תיקייה ראשית. בתוכה:
i. תת תיקייה enums ובתוכה שתי הספריות הקשורות למשימה 1 והן payments ו- colors.
ii. תת תיקייה employee ובתוכה כל המחלקות שקשורות למשימה 2.
iii. תת תיקייה carDealership ובתוכה כל המחלקות וה-enum שקשורות למשימה 3.
iv. קובץ PDF בשם ex4answers עם התשובות למשימה 4.
(יש לשים לב ששם הספרייה שבה שומרים את המחלקות, יהיה תואם לשם החבילה (package) שבה הגדרתם את המחלקות).
את התיקייה הראשית יש לכווץ ולהעלות למודל.

6. בתחילת כל קובץ יש להוסיף את התיעוד הבא. יש לשנות את השם לשם שלכם ואת תעודת הזהות לתעודת הזהות שלכם.

// Assignment: 4

// Author: Israel Israeli, ID: 01234567

7. לאחר שהעליתם את הקובץ שלכם למודל, הורידו אותם מהמודל למחשב שלכם וודאו כי הקובץ תקין וכי העליתם את הקובץ הנכון והמלא. לאחר תום מועד ההגשה לא יתקבלו ערעורים על כך שהעליתם קבצים לא תקינים או שהעליתם בטעות קבצים אחרים / לא נכונים.

שאלות

8. שאלות בנוגע העבודה יש לשאול בפורום באתר המודל של הקורס או בשעות קבלה של המתרגל/ת האחראי/ת בלבד. אין לשלוח שאלות במייל לא למתרגלת האחראית ולא למתרגלים/מרצים אחרים.

9. ניתן לשאול שאלות הבהרה ומיקוד על המשימות שבעבודה במידה ומשימה מסוימת לא ברורה. לא ניתן לשאול על הפתרונות שלכם. לדוגמא, לא ניתן לשאול האם הפתרון שלי נכון, לא ניתן לשאול למה הפתרון לא עובד, וכדומה.

קוד

10. ניתן להשתמש בכל החומר שלמדנו עד למועד פרסום העבודה. בפרט, החומר עוסק וכולל את כל החומר שלמדנו עד לחריגות וקבצים. אך לא כולל רשימות, תורים ומחסניות.

11. הדבר החשוב ביותר הוא שהקוד שלכם יעבוד באופן תקין / נכון. כמו כן, חשוב שהקוד יהיה בנוי בצורה התואמת את פרדיגמת התכנות מונחה העצמים כפי שלמדנו עד כה.

12. יש להקפיד על תכנות נכון. יש לתעד את הקוד באופן מלא בפורמט Javadoc בלבד ובאנגלית בלבד. יש להקפיד על הזחות, שמות משמעותיים וכיוצא באלו.

שונות

13. המשימות בעבודה זו הינן שוות משקל. כלומר, משקל כל משימה הוא 100 חלקי מספר המשימות בעבודה.

14. אם לא צוין במפורש אחרת במשימה ספציפית, בכל המשימות בעבודה זו הניחו כי הקלט מקיים את כל ההנחות הכתובות במשימה. כלומר, אינכם נדרשים לבדוק שהקלט מקיים את ההנחות, אלא אם כן צוין במפורש במשימה שצריך.

15. באפשרותכם להשתמש בשיטות של הספרייה הסטנדרטית במידה ואינן פותרות את המשימה כולה/רובה או חלק גדול ממנה. במידה ואתם בספק, ניתן להעלות את השאלה בפורום העבודות.

בהצלחה!

עבודת בית 4

מחלקות מיוחדות, חבילות, חריגות וקבצים

משימה 1 : אנומרציות – אמצעי תשלום וצבעים (enums)

סעיף א'

תזכורת: כל המחלקות במשימה זו צריך לשמור בספריה payments תחת התיקה enums.

בסעיף זה עליכם ליצור מחלקה המייצגת אמצעי תשלום, מחלקת התשלום מספקת דרך ליצור ולנהל תשלומים, כולל הפקת מספרי אסמכתא והדפסת פרטי תשלום.

עליכם ליצור enum בשם PaymentMethod המכיל את שיטות התשלום הבאות: CREDIT_CARD, DEBIT_CARD, BANK_TRANSFER, PAYPAL and CASH.

בנוסף עליכם ליצור מחלקה בשם Payment. למחלקה זו יהיו שלושה שדות והם:

- amount המייצג את סכום התשלום.
- method המייצג שיטת התשלום.
- referenceNumber המייצג מספר הסימוכין לתשלום.

בנו בנאי המתאים לחתימה הבאה:

```
public Payment(double amount, PaymentMethod method)
```

הבנאי צריך לאתחל את השדות כאשר:

- השדה amount מייצג את סכום התשלום מסוג double.
- השדה method המייצג שיטת התשלום (מופע של ה- PaymentMethod enum).
- השדה referenceNumber המייצג מספר הסימוכין לתשלום מסוג String ומכיל מחרוזת המתחילה ב-"REF" ומסתיימת ב-6 ספרות אקראיות שנגריל.

מחרוזות לדוגמא שהשדה referenceNumber יכול להכיל :

```
"REF361683"  
"REF253377"  
"REF713433"  
"REF369950"  
"REF605862"  
"REF379036"
```

כתבו שיטה printPaymentDetails במחלקה Payment המדפיסה את פרטי התשלום והם: הסכום, אמצעי התשלום ומספר האסמכתא.

לבסוף, הגדירו מחלקה PaymentTest שיוצרת שלושה מופעים של מחלקה Payment עם שיטות תשלום וסכומים שונים, וקוראת לשיטת printPaymentDetails עבור כל מופע.

סעיף ב'

תזכורת: כל המחלקות במשימה זו צריך לשמור בספריה colors תחת התיקיה enums.

בסעיף זה עליכם ליצור מחלקה המייצגת פלטת צבעים.

עליכם ליצור enum בשם Color המכיל את הצבעים הבאים עם ערכי RGB בדיוק כפי שמופיעים ליד כל צבע:

RED (255, 0, 0)
GREEN (0, 255, 0)
BLUE (0, 0, 255)
WHITE (255, 255, 255)
BLACK (0, 0, 0)

בנוסף עליכם ליצור מחלקה בשם ColorPalette. למחלקה זו שדה יחיד:
• colors המייצג מערך צבעים (אובייקטים מסוג Color).

בנו בנאי המקבל כפרמטר מערך של צבעים (אובייקטים מסוג Color) ומאתחל את השדה colors.

כתבו שיטה getColorByName במחלקה ColorPalette שמקבלת ארגומנט String המייצג שם של צבע ומחזירה את האובייקט Color המתאים במידה והוא קיים בפלטת הצבעים. אחרת, אם לא קיים צבע כזה בפלטה, השיטה צריכה להחזיר null.

לבסוף, הגדירו מחלקה ColorPaletteTest שיוצרת מערך של אובייקטי Color עם הערכים של Color enum ומעבירה אותו לבנאי של המחלקה ColorPalette. לאחר מכן, הוא קורא לשיטת getColorByName עבור כל צבע במערך ומדפיס את ערכי ה-RGB של כל צבע.

משימה 2 : records – עובדים (employee)

תזכורת: כל המחלקות במשימה זו צריך לשמור בספרייה employee.

עליכם ליצור את החבילה employee הכוללת ארבע רשומות Java, שהן :
AttendanceRecord, PayrollRecord, LeaveRecord, EmployeeRecord ו-.

כל רשומה תכיל תכונות ספציפיות הקשורות לרשומה זו ותדרוס את השיטה toString כדי לספק ייצוג מחרוזת של תכונות הרשומה. כל רשומה צריכה להכיל תכונות הקשורות לרשומה זו, רשומת העובד צריכה לכלול זיהוי עובד, הייחודי לכל עובד.

EmployeeRecord מייצגת רשומת שכר עם מאפיינים כגון מזהה ID, שם name, גיל age, מחלקה department ויעוד designation.

PayrollRecord מייצגת רשומת עובד עם מאפיינים כגון מזהה ID, שכר בסיסי salary, בONUS bonus ומס tax. בנוסף הוא מכיל:

- שיטה לחישוב השכר הכולל וייצוגו במחרוזת של הרשומה (toString). השכר מחושב כסכום של השכר הבסיסי והבONUS פחות המס.

LeaveRecord מייצגת רשומת חופשה עם מאפיינים כגון מזהה ID, כלל ימי החופש totalLeaves וכמות ימי החופשה שנלקחו leavesTaken. בנוסף הוא מכיל:

- שיטה לחישוב ימי חופשה נותרים וייצוגם במחרוזת של הרשומה (toString). ימי החופשה הנותרים מחושבים כתוצאת הפחתת הימים שנלקחו מסך ימי החופש.

AttendanceRecord מייצגת רשומת נוכחות עם מאפיינים כגון מזהה ID, סך ימי העבודה totalWorkingDays וסך ימי הנוכחות daysPresent. בנוסף הוא מכיל:

- שיטה לחישוב אחוז הנוכחות וייצוגו במחרוזת של הרשומה (toString). אחוז הנוכחות על ידי חלוקת ימי הנוכחות בסך ימי העבודה והכפלה ב-100.

נגדיר מחלקה EmployeeManagementSystem היוצרת מופעים של שתי רשומות מכל סוג (סה"כ 8 מופעים) עם פרטים שונים ולאחר מכן נציג את המידע עבור כל רשומה בעזרת המתודה toString שדרסנו.

משימה 3 : סוכנות רכב

בתרגיל זה אתם מתבקשים לבנות תוכנה המדמה מערכת המנהלת סוכנות רכבים. כל הרכבים שנמכרים בסוכנות הם רכבים ששנת היצור שלהם היא לפחות 2017.

לכל רכב במגרש יש את הנתונים הבאים:

- מספר רכב (מספר בן 6 ספרות)
- שנת ייצור (מספר בין 2017-2023)
- שם היצרן
- קילומטראז'
- מחיר

עליכם לממש את המחלקות הבאות:

1. **מחלקת רכב (Car)** המוגדרת על פי המאפיינים שפורטו לעיל.

המחלקה תכיל את השיטות הבאות:

- **בנאי המקבל את כל הפרמטרים** – בבנאי עליכם לבדוק תקינות הפרמטרים (מס' ספרות במספר הרכב, שנת ייצור גבוהה מ-2017, מחיר/קילומטראז' חיובי, שם יצרן מכיל ערך) ובמידת הצורך תיזרק חריגה עם הודעה מתאימה.
 - לדוגמא: המחיר חייב להיות חיובי, בכל מקרה שבו המחיר יהיה אי-חיובי תיזרק חריגה שתציין שהמחיר חייב להיות חיובי.
- **toString** – מחזרת עם פרטי הרכב כאשר בין כל נתון יש רווח אחד בלבד.
- **שיטה המחשבת הורדת ערך לרכב** – השיטה תקבל את אחוזי ההנחה במחיר ובהתאם לכך תחשב ותעדכן את הערך שלאחר הורדת אותו אחוז ממחירו המקורי של הרכב. במידה והפרמטר שהתקבל שלילי נזרוק חריגה. בנוסף, אם ההורדה במחיר אמורה להיות גבוהה מ-5000 ש"ח לא נבצע את עדכון המחיר ונזרוק חריגה המציינת שהורדת הערך גבוה מדי.
- **דוגמא:** נניח מחיר הרכב 20 אלף ואנו מעוניינים להוריד את העלות ב-2%. כלומר רוצים להוריד 400 ש"ח. הורדה זו תקינה על פי המגבלות הכתובים. לעומת זאת, אם נרצה להוריד את המחיר ב-26% (הורדה של יותר מ-5000) אז תיזרק חריגה.
- **מכירת רכב** – השיטה תכתוב לקובץ המתעד את כל הרכבים שנמכרו. השיטה תשמור את פרטי הרכב בקובץ על פי סדר הנתונים המתואר למעלה. כאשר בין הנתונים יש רווח אחד בלבד.

2. **מחלקת עובד (Employee)** המיישמת את הממשק [Comparable](#).

עובד מוגדר על ידי המאפיינים הבאים:

- שם העובד (שם פרטי בלבד)
- ת.ז.
- מספר מכירות

לכל עובד נוסף את השיטות הבאות:

- **בנאי שמקבל את כל הפרמטרים** – המקבל את שם העובד, ת.ז שלו ומספר המכירות. הבנאי יבדוק את תקינות הנתונים ובכל צורך תיזרק חריגה עם הודעה מתאימה. שם העובד חייב להיות מורכב רק מאותיות, מספר ת.ז יהיה בעל 9 ספרות בדיוק ומספר המכירות יהיה מספר אי שלילי (גדול או שווה ל-0)
- הבנאי יבדוק את תקינות הנתונים ובמידה ואחד הנתונים אינו תקין תיזרק חריגה עם הודעה מתאימה.
- **מכירת רכב** – השיטה תקבל אובייקט מסוג רכב ונתיב לקובץ/אובייקט הכותב לקובץ, בהתאם למימוש (IO/NIO), המתעד את כל הרכבים שנמכרו. השיטה תעדכן את פרטי המכירה המלאים לקובץ (פרטי רכב ופרטי מוכר הרכב). בנוסף, השיטה תעדכן את מספר

- המכירות של העובד ותמכור את הרכב ללקוח (שימו לב עליכם להפעיל את השיטה של מכירת רכב של מחלקת רכב).
- חישוב שכר – השיטה תחשב ותחזיר את השכר של העובד. שכר הבסיס של כל העובדים הוא 6000. ועבור כל מכירה שהם ביצעו הם מקבלים תוספת שכר של 100 ש"ח.
- toString – המחרוזת המשורשרת תכלול את כל פרטי העובד ובנוסף את השכר שלו.
- compareTo(Employee other) (שיטה המיישמת את הממשק Comparable) – השיטה תשווה בין שני עובדים ותחזיר 0 במידה ומספר המכירות שלהם זהה, 1- אם מספר המכירות של העובד האחר גדול יותר. אחרת תחזיר 1.

3. מחלקת סוכנות רכבים (CarDealership)

- במחלקה הזו נממש שיטות גלובליות (public static)
- שיטה גנרית המקבלת ArrayList של אובייקטים (מסוג גנרי). השיטה תמייין את המערך מהגדול לקטן בעזרת פונקציית compareTo.
- תוכנית ראשית (main):
בתוכנית הראשית ניצור-
 - יצירת נתיב לקובץ בשם Sold.txt אליו נכתוב את פרטי הרכבים שנמכרו. קובץ זה מתעד את כל הרכבים שנמכרו.
 - מערך דינאמי (ArrayList) של רכבים (Car) שנאתחל על פי הנתונים הכתובים בקובץ בשם CarDealership.txt. אם אחד הנתונים אודות הרכב לא נכון, לא נסיף רכב זה למערך הדינאמי.
 - מערך דינאמי (ArrayList) של עובדים שנאתחל על פי הנתונים הכתובים בקובץ בשם Employee.txt. אם אחד הנתונים אודות העובד לא נכון, לא נסיף עובד זה למערך הדינאמי.

לאחר מכן, נריץ תפריט (עליכם להיעזר ב- enum):

1. הצגת רשימת העובדים בסוכנות.
2. הצגת הרכבים בסוכנות שטרם נמכרו.
3. מכירת רכב.
4. הוספת רכב לסוכנות.
5. סיום התוכנית.

אופציה 1 : הצגת רשימת העובדים בסוכנות –
 התוכנית תציג את כל העובדים בסוכנות כך שסדר ההדפסות יהיה מהעובד עם מספר מכירות הגבוהה ביותר למספר העובד עם מספר המכירות הנמוך ביותר.
 דוגמא:

Employee name - Israel Israeli, Employee ID-310000000, Sales number -15

Employee name - David Davidi, Employee ID-310000001, Sales number -10

Employee name – Meir Meiri, Employee ID-310000002, Sales number -7

אופציה 2 : הצגת הרכבים בסוכנות שטרם נמכרו.
 התוכנית תציג את פרטי כל הרכבים בסוכנות שטרם מכרו אותם.

List of unsold cars:

Car Number: 123456, Manufacturing Year: 2021,
 Manufacturer: Toyota, Mileage: 15000, Price: 25000.

Car Number: 987654, Manufacturing Year: 2018,
Manufacturer: Honda, Mileage: 20000, Price: 22000.

אופציה 3: מכירת רכב –
תחילה תוצג בפני המשתמש רשימת העובדים (שם + ת.ז שלהם) לאחר מכן המשתמש
יבחר עובד על ידי הקלדת ת"ז שלו.
במידה ות"ז שהמשתמש הקליד לא קיימת תיזרק חריגה שהטיפול בה יהיה קליטה חוזרת.
שימו לב שכמות הפעמים שהמשתמש יכול להזין ת.ז לא תקין היא בלתי מוגבלת.
דוגמא :

Employee name - Israel Israeli, Employee ID-310000000

Employee name - David Davidi, Employee ID-310000001

Employee name – Meir Meiri, Employee ID-310000002

לאחר בחירת העובד התוכנית תציג למשתמש את רשימת הרכבים שטרם נמכרו (נדפס את
כל פרטי הרכב). המשתמש יתבקש לבחור רכב על ידי הקלדת מספר הרכב. גם כן הגישה
תהיה זהה לחוקיות ת"ז של העובד. כלומר, אם לא תקין, תיזרק חריגה והטיפול בה יהיה ע"י בקשה
להזנת פרטים חוזרת.
דוגמא:

List of unsold cars:

Car Number: 123456, Manufacturing Year: 2021,
Manufacturer: Toyota, Mileage: 15000, Price: 25000.

Car Number: 987654, Manufacturing Year: 2018,
Manufacturer: Honda, Mileage: 20000, Price: 22000.

לאחר בחירת העובד והרכב נבצע מכירה של הרכב. מכירת הרכב תכלול עדכון של המספר
המכירות של העובד הרלוונטי, שמירת פרטי הרכב בקובץ הרכבים הנמכרים ומחיקת הרכב
ממערך הרכבים.

אופציה 4 : הוספת רכב לסוכנות-
המשתמש יכניס את פרטי הרכב (מספר רכב, שנת ייצור, שם היצרן, קילומטראז' ומחירו).
נוסיף את הרכב למערך הרכבים אם ניתן. כלומר אם הבנאי זרק חריגה מכון שאחד מפרטי
הרכב לא תקינים נודיע על כך למשתמש ונמשיך בתוכנית.
דוגמא:

Adding a new car:

Enter Car Number (6 digits): 123456

Enter Manufacturing Year (between 2017 and current year): 2022

Enter Manufacturer: Ford

Enter Mileage: 5000

Enter Price: 28000

Car added successfully!

אופציה 5 : סיום התוכנית-
בסיום התוכנית, נשמור את פרטי הרכבים שטרם נמכרו בקובץ CarDealership.txt.
דוגמא:

123456 2022 Ford 5000 28000

987654 2018 Toyota 20000 22000
567890 2020 Honda 15000 18000

נקודות חשובות:

- אסור להוסיף משתנים למחלקות רכב/עובד. ניתן להוסיף פונקציות עזר ושיטות set/get הכרחיות.
- שיטות העזר יהיו בהרשאת private.
- הקבצים המכילים מידע על הרכבים בסוכנות פרטי העובדים יכולים לכלול מידע לא תקין. כלומר, לדוגמא ת.ז של עובד יכול להיות מורכב מ8 ספרות ולא 9. במידה ולא ניתן ליצור אובייקט בגלל שהנתונים לא חוקיים רק את האובייקט הזה לא ניצור. נדפיס "אזהרה" למסך ונמשיך ביצירת האובייקטים האחרים שבאים לאחר מכן.
- בכל פעם שנמכר רכב נוסיף את פרטי הרכב לסוף הקובץ ששומר מידע על רכבים שנמכרו!
- בסיום התוכנית, נשמור את פרטי הרכבים שטרם נמכרו בקובץ המיועד לכך. שימו לב שפרטי הרכבים שמופיעים שם תחילה הם לא בהכרח רלוונטיים בסיום התוכנית ולכן צריך לדרוס את הנתונים המופיעים בו.
- הקבצים המכילים את פרטי העובדים והרכבים בסוכנות מצורפים לעבודה.

משימה 4 : שאלות פתוחות

יש לענות על שאלות הבאות ולהגיש קובץ PDF (בלבד) בשם ex3answers עם התשובות:

1. צפו את הפלט:

```
enum Color {
    RED,
    GREEN,
    BLUE;
    private Color() {
        System.out.println("Constructor called for : "
            + this.toString());
    }

    public void colorInfo() {
        System.out.println("Universal Color");
    }
}

public class Test {
    public static void main(String[] args) {
        Color c1 = Color.RED;
        System.out.println(c1);
        c1.colorInfo();
    }
}
```

2. צפו את הפלט:

```

class Parent {
    public void Print()
    {
        System.out.println("Parent");
    }
}

class Child extends Parent {
    public void Print()
    {
        System.out.println("Child");
    }
}

class Main {
    public static void PrintMain(Parent o)
    {
        o.Print();
    }

    public static void main(String[] args)
    {
        Parent x = new Parent();
        Parent y = new Child();
        Child z = new Child();
        PrintMain(x);
        PrintMain(y);
        PrintMain(z);
    }
}
  
```

a. Child, Child, Child

b. Parent, Child, Child

c. Parent, Child, Parent

d. Parent, Parent, Child

3. מהן רשומות בjava? מה ההבדל ביניהן לבין מחלקות רגילות ומתי נעדיף להשתמש ברשומה על פני מחלקה רגילה?

4. מהן השיטות (מתודות) שמתקבלות אוטומטית כאשר משתמשים ברשומה?

5. מהו enum וכיצד הוא משפר את בהירות ובטיחות הקוד בהשוואה לקבועים רגילים?

6. מהי מטרתם של החבילות (package) בjava? מה הסיבה שמחלקות מאוגדות תחת אותה החבילה?

7. כיצד מצהירים על חבילה בjava? כתבו קוד קצר ובו הצהירו על החבילה example כתת ספרייה שנמצאת בתוך חבילה אחרת ששמה com.

בהצלחה