

מטלה 2 - מטוסים ורכבות

הנחיות כלליות:

- קראו היטב את השאלות והקפידו שהתכנית שלכם פועלו בהתאם לנדרש
- את מטלה יש לפתור בזוגות
- את התוכנית (כל המחלקות) יש לכתוב בתוך **package Ex10_6** כך שיופיע כשורה ראשונה בכל אחד מקבצי המחלקות:

package Ex10_6;

- יש להגיש את כל קבצי ה-JAVA – 4 מחלקות וממשק אחד (**ללא** מחלקת בדיקות) יחד בקובץ דחוס zip אחד.
- יש לדחוס compress **רק** את קבצי ה-JAVA הרלוונטים ישירות (**ולא** תיקיות)

שם הקובץ zip צריך להיות בפורמט הבא כך ששמות הסטודנטים **באנגלית**:

Matala2_[firstname1]_[lastname1]_[firstname2]_[lastname2].zip

לדוגמא:

Matala2_Jerry_Seinfeld_Larry_David.zip

- כדי לוודא את נכונות הפתרונות, בכל תרגיל הריצו את תכניתכם עם מגוון קלטים שונים, כאלה שמופיעים בתוכן התרגיל וקלטים נוספים עליהם חשבתם (וודאו כי הפלט נכון וכי התוכנית אינה קורסת)
- היות ובדיקת התרגילים עשויה להיות אוטומטית, **יש להקפיד על פלטים מדויקים על פי הדוגמאות בתרגיל עד לרמת הרווח**
- על הקוד של כל תרגיל לעבוד ולספק את התוצאה הדרושה עבור קלט שיוזן במשתנים

מטלה 2 - מטוסים ורכבות

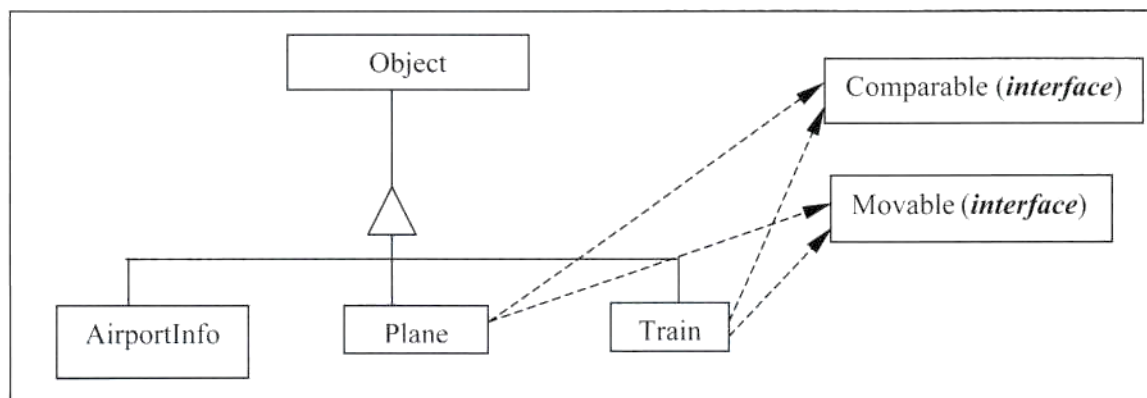
חלק א

רקע

בפרויקט זה עליכם להשתמש בממשק Comparable שהינו חלק מ JAVA API. כדי לדעת אילו שיטות עליכם לכתוב על מנת לממש את הממשק Comparable, עיינו בפירוט שלו המופיע Java API.

מה עליכם לעשות?

עליכם לכתוב תוכנית שתאפשר לדמות מערכת מידע של נמל תעופה. מערכת המידע תאפשר למשתמש לקבל מידע אודות תנועת המטוסים והרכבות בנמל. יחסי הטיפוסים המשתתפים בתוכנית מתוארים בתרשים הבא:



עליכם לשנות את התרשים המופיע בתרשים הנ"ל לפי מה שמוגדר בהמשך התרגיל ובלבד שיחסי הטיפוסים המתוארים בו יישמרו.

בסיום תוכלו לבדוק את התוכנית על ידי המחלקה **TestAirportInfo** שמצורפת כקובץ JAVA למטלה. מחלקה זו כוללת את השיטה `main(...)`, והיא משתמשת במחלקות שכתבתם. אין לצרפה לקובץ ה zip.

הגדרת פעולת השוואה בין עצמים – "מפתח השוואה"

השוואה בין העצמים נעשית על פי הקריטריונים הבאים:

- מטוסים – על פי גובה מרבי לטיסה.
- רכבות – על פי מספר הנוסעים המרבי ברכבת.

עליכם לממש את: Movable הממשק

מגדיר טיפוס של אובייקט היכול לנוע ממקום למקום. פירוט הממשק:

תיאור השיטה	חתימת השיטה
מחזירה את נקודת המוצא של כלי התחבורה	String getSource()
מחזירה את היעד של כלי התחבורה	String getDestination()
מחליפה בין המוצא של כלי התחבורה ליעד שלו	void move()
מחזירה את שם הסוג של כלי התחבורה	String getType()
מחזירה פרטים על אודות כלי התחבורה (ראו פורמט בהמשך)	String details()

עליכם לממש את המחלקות הבאות

1) המחלקה Vehicle

מחלקה זו מגדירה "כלי תחבורה"

מממשת את הממשקים Comparable ו-Moveable.

תכונות:

id - מספר הרישוי של כלי הרכב מסוג מספר שלם

source המוצא מסוג מחרוזת

destination היעד מסוג מחרוזת

פעולות:

Vehicle(int id, String Source, String Destination) פעולה בונה.

בנוסף המחלקה מממשת את הפעולות שהיא יכולה לממש מהממשקים על פי מנגנוני OOP.

2) המחלקה Plane

מחלקה זו מגדירה אובייקט מטיפוס "מטוס" סוג של כלי תחבורה.

תכונות:

height גובה המרבי לטיסה מסוג מספר שלם.

פעולות:

Plane(String originPoint, String finalPoint,int id ,int height) פעולה בונה.

getType() מחזירה את שם הסוג של כלי התחבורה

compareTo(Object o) (לפי הגובה המרבי לטיסה)

details מחזירה מחרוזת לפי הפורמט הבא:

```
Movable Type: Plane
Number [id]
Source = [source name]
Destination = [destination name]
Max altitude: [max height of plane]
```

3) המחלקה Train

מחלקה זו מגדירה אובייקט מטיפוס "רכבת" סוג של כלי תחבורה.

תכונות:

maxPas מספר הנוסעים המירבי מסוג מספר שלם.

פעולות:

getType() מחזירה את שם הסוג של כלי התחבורה

compareTo(Object o) (לפי מספר הנוסעים המרבי ברכבת)

details שתחזיר מחרוזת לפי הפורמט הבא:

```
Movable Type: Train
Number [id]
Source = [source name]
Destination = [destination name]
Passengers: [max passengers count]
```

4) המחלקה AirportInfo

זו מחלקת שירות הכוללת פעולות מחלקה המאפשרות למיין כלי תחבורה מסוגים שונים, ולהדפיס דוחות על תנועתם בנמל. עליכם לממש את הממשק הבא:

תיאור השיטה	חתימת השיטה
ממיינת מערך של אובייקטים מטיפוס Comparable לפי "מפתח ההשוואה" שהוגדר בעבורם	static void sortTransport (Comparable[] transport)
מחזירה את הדוחות של כל האובייקטים מטיפוס Movable, שהיא מקבלת (ראו פורמט בהמשך, בפלט של הרצת תוכנית הבדיקה)	static String reportAll (Movable[] movable)

כדי לממש את sortTransport עליכם להשתמש באחד מאלגוריתמים המיון שלמדתם (מיון בועות או מיון בחירה).

השיטה reportAll(...) תחזיר מחרוזת הכוללת את הדוחות של כל האובייקטים (ראו פורמט בהמשך, בפלט של הרצת תוכנית הבדיקה).

הרצת הפרויקט

כדי להריץ את הפרויקט ולבדוק את המחלקות שכתבתם צרפו אל הפרויקט את המחלקה TestAirportInfo, שכתבתי בעבורכם ומצורפת למטלה.

עיינו היטב בקוד. המחלקה כוללת את השיטה main(...), והיא תאפשר לכם לבדוק את המחלקות שכתבתם. לאחר מכן הדרו (קמפלו) את הפרויקט הכולל את המחלקה והריצו את התוכנית.

דוגמא לאחד הפלטים (לפי אקראיות) , צריך לצאת **בדיוק** לפי הפורמט הבא :

```
Movable Type: Train
Number 54
Source = London
Destination = Copenhagen
Passengers: 18
-----
```

```
Movable Type: Train
Number 180
Source = Roma
Destination = Barcelona
Passengers: 39
-----
```

```
Movable Type: Train
Number 159
Source = New-York
Destination = Bordeaux
Passengers: 41
-----
```

```
Movable Type: Train
Number 265
Source = London
Destination = Helsinki
Passengers: 107
-----
```

```
Movable Type: Train
Number 26
Source = New-York
Destination = Helsinki
Passengers: 130
-----
```

```
---Sorting planes...---
```

Movable Type: Plane
 Number 955
 Source = London
 Destination = Copenhagen
 Max altitude: 3

Movable Type: Plane
 Number 1303
 Source = Paris
 Destination = Madrid
 Max altitude: 20

Movable Type: Plane
 Number 659
 Source = Roma
 Destination = Copenhagen
 Max altitude: 30

Movable Type: Plane
 Number 690
 Source = Jerusalem
 Destination = Helsinki
 Max altitude: 32

Movable Type: Plane
 Number 967
 Source = New-York
 Destination = Helsinki
 Max altitude: 63

---All together...---

Movable Type: Train
 Number 54
 Source = London
 Destination = Copenhagen
 Passengers: 18

Movable Type: Plane
 Number 955
 Source = London
 Destination = Copenhagen

Max altitude: 3

Movable Type: Train

Number 180

Source = Roma

Destination = Barcelona

Passengers: 39

Movable Type: Plane

Number 1303

Source = Paris

Destination = Madrid

Max altitude: 20

Movable Type: Train

Number 159

Source = New-York

Destination = Bordeaux

Passengers: 41

Movable Type: Plane

Number 659

Source = Roma

Destination = Copenhagen

Max altitude: 30

Movable Type: Train

Number 265

Source = London

Destination = Helsinki

Passengers: 107

Movable Type: Plane

Number 690

Source = Jerusalem

Destination = Helsinki

Max altitude: 32

Movable Type: Train

Number 26

Source = New-York

Destination = Helsinki

Passengers: 130

Movable Type: Plane
Number 967
Source = New-York
Destination = Helsinki
Max altitude: 63

---All together after move...---

Movable Type: Train
Number 54
Source = Copenhagen
Destination = London
Passengers: 18

Movable Type: Plane
Number 955
Source = Copenhagen
Destination = London
Max altitude: 3

Movable Type: Train
Number 180
Source = Barcelona
Destination = Roma
Passengers: 39

Movable Type: Plane
Number 1303
Source = Madrid
Destination = Paris
Max altitude: 20

Movable Type: Train
Number 159
Source = Bordeaux
Destination = New-York
Passengers: 41

Movable Type: Plane
Number 659
Source = Copenhagen
Destination = Roma
Max altitude: 30

Movable Type: Train
 Number 265
 Source = Helsinki
 Destination = London
 Passengers: 107

Movable Type: Plane
 Number 690
 Source = Helsinki
 Destination = Jerusalem
 Max altitude: 32

Movable Type: Train
 Number 26
 Source = Helsinki
 Destination = New-York
 Passengers: 130

Movable Type: Plane
 Number 967
 Source = Helsinki
 Destination = New-York
 Max altitude: 63

חלק ב

שאלות מילוליות

יש לענות בכתב ולהעלות פתרונות בקובץ PDF בנוסף מחוץ לקובץ zip

1. באילו ממשקים השתמשתם בתרגיל זה? מה תפקידם בתוכנית?
 2. איזה עיקרון של OOP יכולתם להפעיל משום שמימשתם את הממשקים?
 3. האם הייתם יכולים להשיג אותה תוצאה על ידי שימוש במנגנון ירושה או במחלקות מופשטות? אם כן, הסבירו כיצד. אם לא, הסבירו מדוע.
 4. מה היה קורה אילו במחלקה Plane הייתם מממשים רק את השיטה details() מתוך הממשק Moveable? האם הייתם מצליחים להדר את התוכנית? הסבירו.
- בשאלות הבאות סמנו את התשובה הנכונה (רק תשובה אחת נכונה):

5. מהו ממשק (interface)?

- א. טיפוס המהווה אוסף של הצהרות על קבועים ושיטות.
 - ב. אוסף של שיטות פומביות של מחלקה.
 - ג. מחלקה שתת-מחלקה יכולה לפרט (extends).
 - ד. מנגנון הקובע היררכיה בין מחלקות.
6. האם ממשק יכול להכיל מימוש של שיטה?
- א. לא.
 - ב. לפעמים.
 - ג. כן.
 - ד. תמיד.

7. האם ניתן להשתמש בשם של ממשק (interface) כדי להגדיר טיפוס משתנה. לדוגמה כך:

```
public static void main(String[] args) {
    SomeInterface x;
}
```

- א. לא, משתנה חייב להיות מטיפוס בסיסי כמו *int*, *double* או *boolean*.
- ב. לא, משתנה חייב להיות מוגדר כטיפוס בסיסי או כהפניה לטיפוס של אובייקט.
- ג. כן, המשתנה x יכול להיות מוגדר כהפניה לכל אובייקט שהמחלקה שלו מממשת את הממשק.
- ד. לא, משתנה חייב להיות מוגדר כהפניה לטיפוס של אובייקט.

בהצלחה!