

מטלת מנחה (ממ"ן) 13

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידות 5 – 6 נושא המטלה: לולאות ומערכים

מספר השאלות: 1 משקל המטלה: 5 נקודות

סמסטר: 2020 מועד אחרון להגשה: 9.5.2020

(ת)

במטלה זו אנו משתמשים במחלקות Time1 ו-Train שכתבנו בממ"ן 12.

אתם יכולים להשתמש במחלקות Time1 ו-Train שכתבתם או בקבצים Time1.class ו-Train.class שיהיו באתר בתוך יחידה 6 בצמוד למטלה 13. נשים את הקבצים האלו באתר רק אחרי ההגשה של מטלה 12.

אנא קראו את הכתוב במדריך שנמצא ביחידה 1 ב"מדריכי עזר וקישורים" בדף הבית של האתר הקורס. המדריך נקרא:

מדריך עזר קצר לשימוש בקובצי מחלקות (class) הקיימים בפרוייקט שאתם יוצרים ב-BlueJ קובץ pdf

כך תדעו איך להשתמש במחלקה שכבר כתובה, וניתנת לכם כקובץ class ללא הקוד. שמנו באתר טסטר בסיסי לבדיקה ראשונית של המטלה. חובה להריץ את המטלה מול הטסטר ולבדוק שאין טעויות קומפילציה.

שאלה 1 - להרצה (100%)

המחלקה RailwayStation מייצגת תחנת רכבת.

הייצוג נעשה על-ידי מערך ששומר את **לוח הנסיעות** (הרכבות). התכונות במחלקה הן:

- מערך של הרכבות `Train [] _station`
- מספר הרכבות שיוצאות מהתחנה `int _noOfTrs`

כמו כן קיים במחלקה קבוע שלם `MAX_TRAINS` המציין את המספר המקסימלי של רכבות ביממה – 100.

הרכבות (כלומר האובייקטים מהמחלקה Train) נמצאים במערך ברצף, ללא "חורים" מתחילת המערך. המערך צריך להישאר כך (ללא חורים) לאחר כל פעולה.

עליכם לכתוב את המימוש ב-Java של המחלקה **RailwayStation**. מימוש המחלקה כולל את הסעיפים שלהלן:

1. הגדרת הקבועים של המחלקה.
2. הגדרת התכונות של המחלקה.
3. בנאי **ריק** המאתחל את תכונות המחלקה כך שמערך הנסיעות יהיה בגודל מקסימלי.
4. שיטה (**addTrain**) בוליאנית המוסיפה נסיעה ללוח הרכבות. השיטה מקבלת את הנסיעה כפרמטר. השיטה מחזירה ערך **true** אם ההוספה התבצעה כשורה, אם לא, השיטה תחזיר **false**. אם הנסיעה הזו כבר קיימת בלוח הרכבות, או אם אין מקום בלוח הרכבות, השיטה מחזירה **false**. זכרו שלוח הרכבות (המערך) אינו מסודר לפי סדר כלשהו, ולכן ההכנסה של הרכבת החדשה חייבת להיות לתוך התא הראשון שפנוי במערך.
5. שיטה (**removeTrain**) בוליאנית המוחקת נסיעה מלוח הרכבות. השיטה מקבלת את הנסיעה כפרמטר. השיטה מחזירה ערך **true** אם המחיקה התבצעה כשורה, אם לא, השיטה תחזיר **false**. אם הנסיעה אינה בלוח הרכבות, או אם אין נסיעות (המערך ריק), השיטה תחזיר **false**. שימו לב שלאחר המחיקה אסור שישאר תא ריק במערך (חור), ולכן צריך להכניס לתא זה במקום הרכבת שנמחקה את הרכבת האחרונה בלוח הרכבות. **לא להזיז את כל התאים צעד אחד אחורה.**
6. שיטה (**firstDepartureToDestination**) המקבלת עיר כלשהי **place**, מחזירה את הזמן בו יוצאת הנסיעה הראשונה באותו יום לכיוון היעד **place**. אם אין אף נסיעה באותו יום אל היעד **place** או שאין נסיעות (המערך ריק) יוחזר **null**.
7. שיטה (**howManyFullTrains**) המחזירה מספר האומר כמה רכבות מלאות יש באותו יום.
8. שיטה (**mostPopularDestination**) המחזירה את העיר הכי פופולרית באותו יום (כלומר העיר אליה מגיעות הכי הרבה רכבות, לא תלוי במספר הנוסעים). אם יש כמה ערים פופולריות באותה מידה, יש להחזיר את הראשונה שנמצאה. אם אין נסיעות (המערך ריק) יש להחזיר **null**.
9. שיטה (**mostExpensiveTicket**) המחזירה את הנסיעה שהכרטיס שלה הוא היקר ביותר. אם יש כמה נסיעות שהכרטיסים שלהם באותו מחיר יקר ביותר, יש להחזיר את הראשונה שנמצאה. אם אין נסיעות (המערך ריק) יש להחזיר **null**.
10. שיטה (**longestTrain**) המחזירה את הנסיעה הארוכה ביותר במערך הרכבות. אם יש כמה נסיעות שזמן נסיעתן הוא הארוך ביותר, יש להחזיר את הראשונה שנמצאה. אם אין נסיעות (המערך ריק) יש להחזיר **null**.
11. שיטה (**toString**) המחזירה מחרוזת המתארת את כל הרכבות במערך הרכבות כסדרן לפי המערך (אין צורך למיין לפי זמנים או משהו אחר). כל נסיעה תהיה בשורה נפרדת. ובתחילה תהיה כותרת. ראו את הדוגמא הבאה:
The trains today are:
Train to Jerusalem departs at 12:00. Train is full
Train to Tel-Aviv departs at 10:50. Train is full
Train to Eilat departs at 11:35. Train is not full
אם אין נסיעות (המערך ריק) המחרוזת תהיה: **There are no trains today.**

לפניכם רשימת החתימות של הבנאי ושיטות המחלקה:

- `public RailwayStation (String city)`
- `public boolean addTrain(Train f)`
- `public boolean removeTrain(Train f)`
- `public Time1 firstDepartureToDestination (String place)`
- `public String toString()`
- `public int howManyFullTrains()`
- `public String mostPopularDestination()`
- `public Train mostExpensiveTicket()`
- `public Train longestTrain()`

שימו לב לא לבצע aliasing במקומות המועדים.

מותר להוסיף שיטות נוספות (פרטיות), לפי ראות עיניכם.

אתם צריכים לכתוב בעצמכם API למחלקה, לבנאים ולשיטות לפי הנהוג בכתיבת API. כמו כן, עליכם לתעד בתיעוד פנימי כל מה שדורש הבהרה ואינו פשוט.

שימו לב,

באתר הקורס תמצאו גם טסטר לבדיקת האיות והפרמטרים של השמות של השיטות והמחלקה שאתם צריכים לכתוב. חובה עליכם לבדוק את המחלקה שכתבתם בטסטר זה, ולהגיש אותה רק אם הטסטר עובר קומפילציה. שימו לב שהטסטר לא מכסה את כל האפשרויות, ובפרט לא את מקרי הקצה. הוא רק בודק את השמות של השיטות במחלקות כלומר שגיאות קומפילציה. מאד מומלץ להוסיף לו בדיקות.

הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הקפידו לתעד בתיעוד פנימי וב-API את כל השיטות שיש במחלקות השונות.
3. הקפידו ששמות השיטות יהיו בדיוק כפי שכתוב במטלה. וכן שההדפסות יהיו בדיוק כפי שמופיע במטלה.
4. עליכם להגיש את הקובץ `RailwayStation.java`, עטפו אותו בקובץ `zip` ושלו. אין לשלוח קבצים נוספים.

