

## הנדסת תוכנה – תרגיל בית 2

### **דגשים להגשת המטלה**

1. תאריך הגשה: יום חמישי 24.06.2021 בשעה 23:59.
2. הגשה בזוגות בלבד!
3. הקוד חייב להיכתב בהתאם למוסכמות כתיבת הקוד בקורס כולל תיעוד כנדרש. קוד שלא עומד בדרישות יגרור הורדת ניקוד.
4. ההגשה מתבצעת ב-Moodle באזור המיועד על ידי אחד מהשותפים, לאחר יצירת קבוצה.
5. כל איחור בהגשה יגרור הורדת 20% מהציון בכל יום.
6. פורמט הגשת התרגיל נמצא בקובץ ההנחיות ב-Moodle. כל חריגה מפורמט זה תגרור ציון 0.
7. שאלות והבהרות ייכתבו בפורום ייעודי שייפתח לתרגיל הבית ב-Moodle.

### מטרת התרגיל

עבודה עם מחלקות והורשה.

### הכנות טרם תחילת התרגיל

1. פתיחת פרויקט Java חדש.
2. הורדת קבצי התרגיל, והעתקת הקובץ Main.java בלבד אל תוך תיקיית ה-src.

### הוראות כלליות

1. יש לבדוק שהקוד עובר הידור (קומפילציה) ללא שגיאות.
2. מומלץ להריץ את התוכנית עם מספר קלטים שונים ולחשוב על מקרי קצה אפשריים.
3. מומלץ לחזור על התרגולים וההרצאות וכן להיעזר באינטרנט.
4. מומלץ להשתמש ב-Git במהלך כתיבת התרגיל.
5. יש לתעד את המחלקות ואת הפעולות שבהן, לפי מדריך התיעוד שבאתר הקורס.

### הוראות הגשה

1. יש למלא אחר הוראות ההגשה בהתאם למסמך הדרישות "הנחיות כלליות לפתרון והגשת תרגילי הבית" אשר מופיע באתר הקורס.
2. הגשה אלקטרונית בלבד דרך אתר הקורס ב-moodle. ההגשה תכלול את קובץ ה-zip בלבד.
3. ההגשה מתבצעת על ידי אחד מבני הזוג לאחר שיצר קבוצה ובן הזוג השני הצטרף אליה.
4. תרגיל בית שלא יוגש על פי הוראות ההגשה – לא ייבדק.
5. יש להקפיד על יושרת הכנת התרגיל וההגשה.
6. יש לוודא כי הקוד מתקמפל – קוד אשר לא יעבור הידור יקבל ציון 0.
7. אין צורך להגיש את קבצי הקלט והפלט אשר ניתנים כחלק מתרגיל זה.

בתרגיל זה נמדל מערכת לניהול קבצים במחשב.

### סיפור המערכת

המערכת מורכבת מפריטי אחסון שונים. לכל פריט אחסון יש שם, תאריך יצירה (אשר אינו משתנה מרגע יצירת הפריט) וגודל אותו הוא תופס בזיכרון.

ישנם שני סוגים של פריטי אחסון. הסוג הראשון הוא קובץ, אשר מאופיין גם על ידי הסיומת שלו. כל קובץ מכיל מחרוזת, אשר נכנה אותה בתור התוכן של הקובץ. הסוג השני הוא תיקייה. כל תיקייה יכולה להכיל בתוכה מספר לא מוגבל של פריטי אחסון נוספים.

בנוסף, ידוע כי כל פריט אחסון הינו קובץ או תיקייה (ולא שניהם).

### מימוש המערכת

על מנת לממש את המערכת, תצטרכו ליצור מספר מחלקות. מחלקות אלו יתבססו אחת על השנייה, וביחד יהוו את מכלול המערכת.

ראשית, עליכם ליצור enum בשם `SortingField` אשר ייצג את האופן שבו נמיין את פריטי האחסון בעת הצגת (הרחבה בהמשך).

קיימים שלושה אופני מיון אפשריים:

- מיון לפי שמות הפריטים.
- מיון לפי גודל הפריטים.
- מיון לפי תהליך היצירה של הפריטים.

לאחר מכן, עליכם להגדיר מספר מחלקות נוספות.

### מחלקת `StorageItem`

מחלקה זו מייצגת פריט אחסון.

על המחלקה לכלול:

- תכונות מתאימות.
- פעולות `get` ו-`set` לתכונות הנדרשות בכך.
- בנאי המקבל את שם הפריט. על הבנאי לאתחל את השם לשם שהתקבל. בנוסף, על הבנאי לאתחל את תאריך היצירה של הפריט לתאריך רנדומלי. הסבר על אופן הגרלת התאריך מופיע בהמשך המסמך.
- פעולה מופשטת `int getSize()` אשר מחזירה את גודלו של הפריט.
- פעולה `void printTree(SortingField field)` אשר מדפיסה את עץ המערכת החל מן הפריט הנוכחי. הסבר על אופן ההדפסה מופיע בהמשך המסמך.

### מחלקת File

מחלקה זו מייצגת קובץ.

על המחלקה לכלול:

- תכונות מתאימות.
- בנאי אשר מקבל את שם הקובץ ואת הסיומת שלו ומאתחל את תכונות אלו בהתאם. בנוסף, על הבנאי לאתחל את תוכן הקובץ להיות ריק.
- פעולה `String getName()` אשר מחזירה מחרוזת המייצגת את שמו של הקובץ ואת הסיומת שלו בפורמט `name.ext`, כאשר `name` זהו שמו ו-`ext` זו הסיומת שלו.
- פעולה `int getSize()` אשר מחזירה את גודלו של הקובץ. גודל של קובץ מוגדר להיות מספר התווים שבתוכן הקובץ.
- פעולה `void printContent()` אשר מדפיסה את תוכן הקובץ בפורמט הבא:

name Size: sizeMb Created: date

content

כאשר `name` זהו שם הקובץ (עם הסיומת), `size` זהו גודל הקובץ, `date` זהו תאריך היצירה של הקובץ ו-`content` זהו תוכן הקובץ.

### מחלקת Folder

מחלקה זו מייצגת תיקייה.

על המחלקה לכלול:

- תכונות מתאימות.
- פעולות `get` ו-`set` לתכונות הנדרשות בכך.
- בנאי אשר מקבל את שם התיקייה, ומאתחל אותו. בנוסף, על הבנאי לאתחל את רשימת הפריטים שבתקייה לרשימה ריקה.
- פעולה `int getSize()` אשר מחזירה את גודלה של התיקייה. גודל של תיקייה מוגדר להיות סכום הגדלים של כלל פריטי האחסון אשר נמצאים בתוך התיקייה.
- פעולה `boolean addItem(Item item)` אשר מקבלת פריט אחסון ומוסיפה אותו לתיקייה, במידה ולא קיים פריט אחסון נוסף בעל שם זהה בתיקייה. במידה ולא קיים פריט כזה, הפריט יתווסף לתיקייה, והפעולה תחזיר `true`. במידה וכן קיים פריט כזה, הפעולה תחזיר `false` מבלי להוסיף את הפריט.
- הערה: ניתן ליצור באותה תיקייה שני קבצים בעלי אותו שם אך עם סיומת שונה.
- פעולה `File findFile(String path)` אשר מקבלת נתיב לקובץ ומחזירה את הקובץ בעל נתיב זה. במידה ולא קיים קובץ כזה, הפעולה תחזיר `null`. פורמט הנתיב הוא שמות של תיקיות וקבצים (שלאו דווקא קיימים), אשר מופרדים ביניהם על ידי התי "/>

### הגרלת תאריך

בעת יצירת פריט אחסון, עליכם להגריל עבורו את תאריך היצירה שלו. הגרלת התאריך הינה הכרחית, מאחר ולא ניתן להשתמש בזמן האמיתי של יצירת הפריט, שכן שימוש בו יוביל לפלט אחר בכל ריצה של התוכנית.

על מנת להגריל תאריך, נשתמש בחותמות זמן: קידוד של תאריך אל מספר מטיפוס long, אשר ערכו שווה למספר המילישניות שעברו מאז התאריך 01.01.1970 בשעה 00:00:00. שימוש בחותמת זמן יאפשר לנו להגריל מספר בודד ולהמירו לתאריך.

המערכת תומכת בקבצים שנוצרו בין התאריך 01.01.2017 בשעה 00:00:00 לתאריך 31.12.2021 בשעה 23:59:59.

על כן, עליכם להגריל מספר אשר נמצא בטווח שבין חותמות הזמן של שני התאריכים הללו.

כמו בתרגיל בית 1, ההגרלה תתבצע באמצעות המשתנה הסטטי rnd אשר נמצא במחלקת Main שמסופקת לכם כחלק מתרגיל זה.

הכוונות:

- לשם עבודה עם תאריכים וחותמות זמן, קראו על המחלקות Date ו-Timestamp.
- חשבו כיצד להגריל חותמת זמן בין שתי חותמות נתונות על ידי שימוש באופרטור מודולו.

### מיון פריטי האחסון

כפי שהוזכר, ישנן שלוש אפשרויות למיין לפיהן את פריטי האחסון בעת הדפסת עץ הפריטים.

- מיון לפי שמות הפריטים. במיון זה נציג את הפריטים הנמצאים בכל תיקייה לפי סדר לקסיקוגרפי של שמותיהם.
- מיון לפי גודל הפריטים. במיון זה נציג את הפריטים הנמצאים בכל תיקייה לפי סדר עולה של הגודל שלהם. במידה ויש פריטי אחסון בעלי אותו גודל, יש למיין אותם (בתור מיון משני) לפי שמותיהם.
- מיון לפי תאריך היצירה של הפריטים. במיון זה נציג את הפריטים הנמצאים בכל תיקייה לפי סדר עולה של תאריך היצירה שלהם. במידה ויש פריטי אחסון בעלי אותו תאריך יצירה, יש למיין אותם (בתור מיון משני) לפי שמותיהם.

כאשר ממיינים לפי שמות הקבצים (כמיון ראשוני או משני) יש להתייחס לאותיות גדולות וקטנות.

הכוונות:

- על מנת למיין רשימה לפי פעולה מותאמת אישית ניתן להיעזר בפעולה `Comparator.comparing`.
- על מנת לבצע מיון משני על רשימה לפי פעולה מותאמת אישית ניתן להיעזר בפעולה `.thenComparing`.

### הדפסת עץ פריטי האחסון

כאשר מדפיסים את עץ פריטי האחסון, יש להדפיס כל פריט בשורה נפרדת, תוך שמירה על מבנה העץ. כאשר מדפיסים תיקייה, יש להתחיל בלהדפיס את שמה. לאחר מכן, יש להדפיס את כלל פריטי האחסון אשר נמצאים בה, כאשר כל פריט אשר נמצא בתוך התיקייה יודפס לאחר הזחה. ההזחה מורכבת מן התו '|', ולאחריו ארבעה רווחים.

לדוגמה: אם יש תיקייה A ובתוכה יש קובץ aa.py, קובץ Aa.txt ותיקייה נוספת B אשר בה יש קובץ bye.log, ותיקייה נוספת C אשר בה יש קובץ code.java, כאשר נדפיס את עץ הפריטים החל מתיקייה A, כאשר נקבע את המיון להיות לפי שמות הפריטים, נקבל:

```
A
| Aa.txt
| B
| | C
| | | code.java
| | bye.log
| aa.py
```

### הערות והנחות

- יש ליצור כל מחלקה (ואת ה-enum) בקובץ נפרד.
- בכל מחלקה ניתן להוסיף פעולות ותכונות נוספות.
- במהלך כתיבת הקוד, יש לשמור על עקרונות תכנות נכונים, כפי שנלמד בקורס.
- יש לתעד את הפעולות והמחלקות כפי שמופיע במסמך התיעוד שבאתר הקורס.
- בכל המחלקות אשר מקבלות בבנאי את שם פריט האחסון ניתן להניח כי השם מורכב מאותיות (גדולות וקטנות) ומספרים בלבד.

### **הרצת התוכנית וביצוע בדיקות**

בפעולה הראשית קיים קטע קוד אשר משמש לביצוע בדיקת לקוד שכתבתם.

להזכירכם, חלק מן הבדיקה נעשה באופן אוטומטי, ולכן אין לשנות את הפעולה הראשית, ובפרט אין לשנות את פעולות ההדפסה המתבצעות בה.

מצורף לתרגיל זה קובץ הפלט HW2\_output.txt, על מנת שתוכלו לבצע את ההשוואה באופן ידני (או על ידי שימוש ב-DiffMerge).

בהצלחה.