תוכנה 1 – אביב תשע״ג

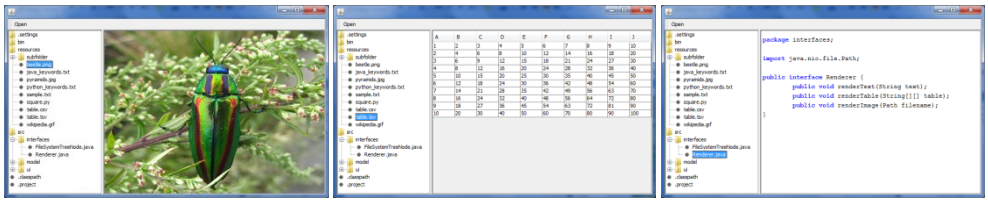
תרגיל מספר 11

<u>הנחיות כלליות:</u>
קראו בעיון את קובץ נהלי הגשת התרגילים אשר נמצא באתר הקורס.

- הגשת התרגיל תעשה במערכת ה moodle בלבד (http://moodle.tau.ac.il/).
- יש להגיש קובץ zip יחיד הנושא את שם המשתמש ומספר התרגיל (לדוגמא, עבור המשתמש zvainer יקרא הקובץ zvainer_hw11.zip). קובץ ה-zip יכיל:
  א. קובץ פרטים אישיים בשם details.txt המכיל את שמכם ומספר ת.ז.
  ב. קבצי ה- java של התוכניות אותם התבקשתם לממש.
  ג. קובץ טקסט אחד עם העתק של כל קבצי ה- java.

<u>תצוגה מקדימה</u>

בתרגיל זה נכתוב מערכת המספקת תצוגה מקדימה לקבצים. המערכת תציג חלון אשר בצידו השמאלי תת-עץ של מערכת הקבצים. לכל קובץ ו/או ספרייה שייבחרו המערכת תייצר תצוגה מקדימה אשר תוצג בצידו הימני של החלון.



<u>הערה:</u> בתרגיל זה יש לממש מבנה נתונים המייצג את מבנה העץ של מערכת הקבצים וכן לייצר תאור של התצוגה המקדימה. אינכם נדרשים לטפל כלל בממשק המשתמש.

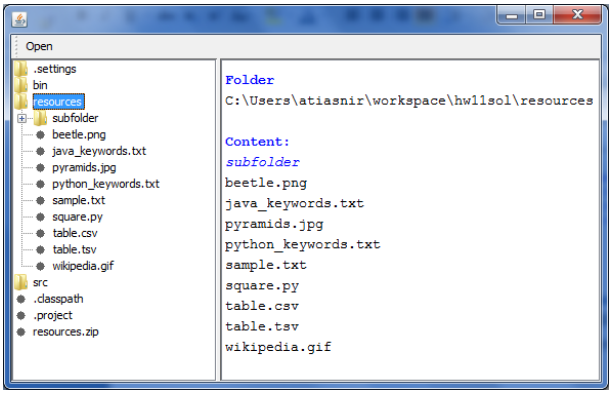
- בקוד השלד הנמצא באתר, אנו מספקים:
  - את המנשק FileSystemTreeNode המתאר צומת בעץ. מנשק זה מרחיב את המנשק TreeNode שהוא מנשק סטנדרטי ב-java. פרטים נוספים עבור מנשק זה בקישור: http://docs.oracle.com/javase/7/docs/api/javax/swing/tree/TreeNode.html
  - את הממשק הגרפי הבנוי ממחלקות הנמצאות בחבילה ui. אין לבצע שינויים במחלקות אשר נמצאות בחבילה זו.
  - את המחלקה NodeFactory המשמשת ליצירת מופעים של צמתים בעץ (מחלקות המממשות את המנשק FileSystemTreeNode).
- כדי ליצור את התצוגה המקדימה יש להיעזר במנשק Renderer המועבר כפרמטר למתודה renderPreview של המנשק FileSystemTreeNode. למנשק זה מתודות ליצירת תצוגה מבוססת טקסט, תצוגה מבוססת טבלה ותצוגה מבוססת תמונה (פרטים בהמשך).

<u>דרישות המערכת:</u>

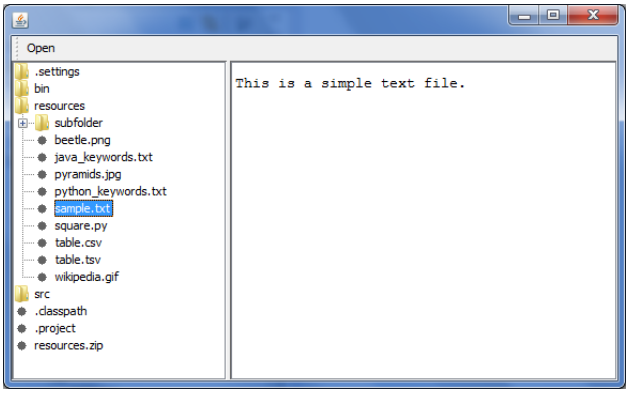
1. המערכת תתחזק מבנה נתונים בצורת עץ של קבצים וספריות. כל צומת בעץ ייוצג ע"י מחלקה המממשת את המנשק (המסופק) FileSystemTreeNode.

2. מספר הקבצים בכל תתי הספריות עלול להיות רב, לדוגמא, לא נרצה לסרוק את כל הדיסק כאשר ספריית הבסיס היא C:\. לכן, יש להקפיד לטעון מידע רק עפ"י דרישה. בפרט, עבור ספרייה נתונה יש לטעון רק את שמות הקבצים ותתי-הספריות הנמצאים ישירות מתחתיה ורק כאשר יש צורך בכך.
3. תצוגה מקדימה לספרייה תכלול את האלמנטים הבאים:
   א. מסלול מלא לספרייה
   ב. שמות תתי הספריות מסודרות לפי סדר א"ב. שם כל ספרייה יופיע בגופן מוטה.
   ג. שמות הקבצים בספרייה מסודרים לפי סדר א"ב.

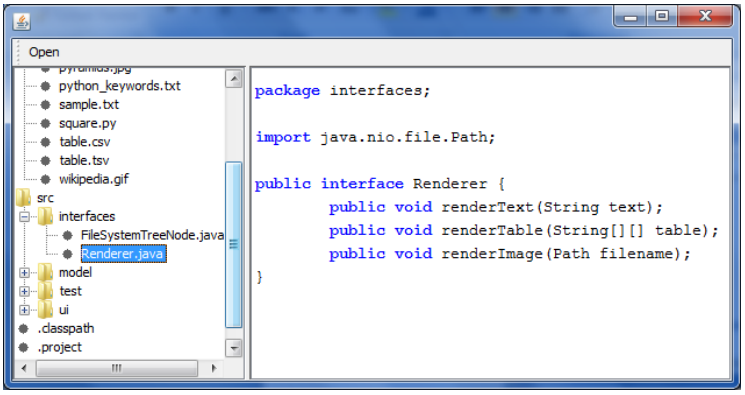
דוגמא לתצוגה מקדימה של הספרייה resources המסופקת עם התרגיל. ספרייה זו כוללת תת-ספריה אחת (subfolder) ומספר קבצים.



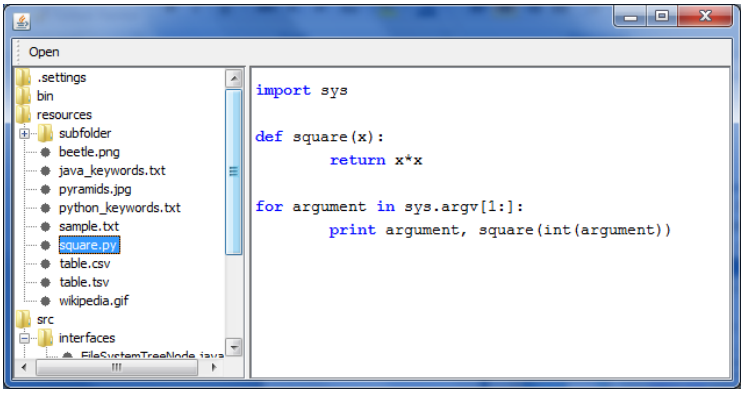
4. הסיומת txt תסמן קבצי טקסט. יש להציג את תוכנן של 100 השורות הראשונות בקובץ.



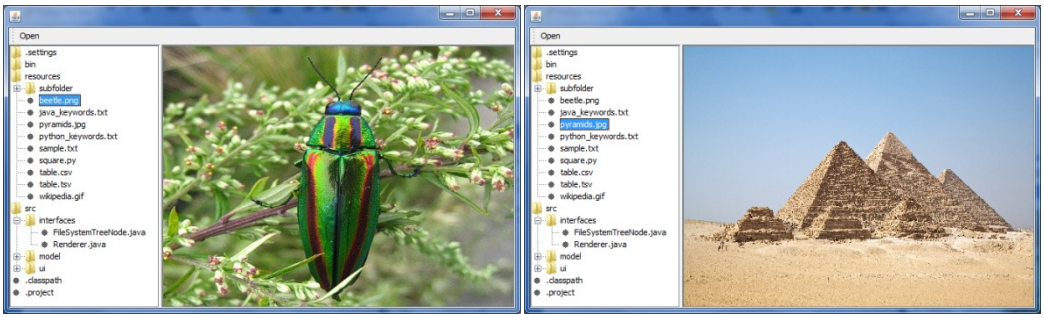
5. הסיומת java תסמן קבצי קוד java. יש להציג את תוכנן של 100 השורות הראשונות בקובץ ולהדגיש את מילות המפתח בשפה. לנוחיותכם הספרייה resources מכילה קובץ בשם java_keywords.txt ובו מילות המפתח.



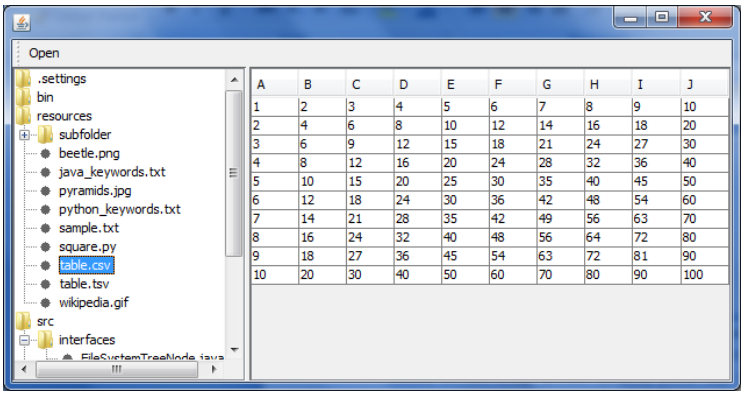
6. הסיומת py תסמן קבצי קוד Python. יש להציג את תוכנן של 100 השורות הראשונות בקובץ ולהדגיש את מילות המפתח בשפה. הספרייה resources מכילה קובץ בשם python_keywords.txt המכיל את מילות המפתח בשפה.



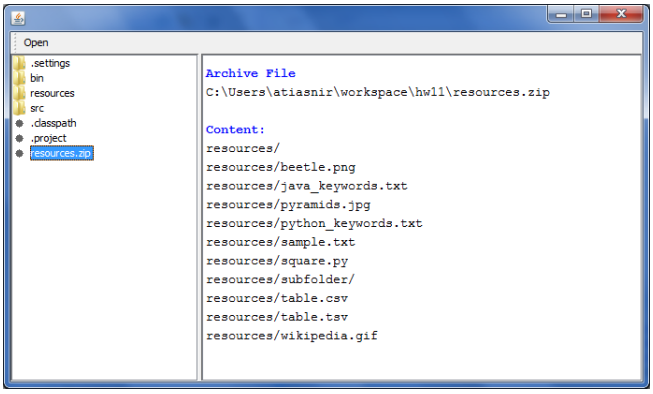
7. קבצים המסתיימים באחת מהסיומות: png, gif, bmp ו-jpg הם קבצי תמונה.



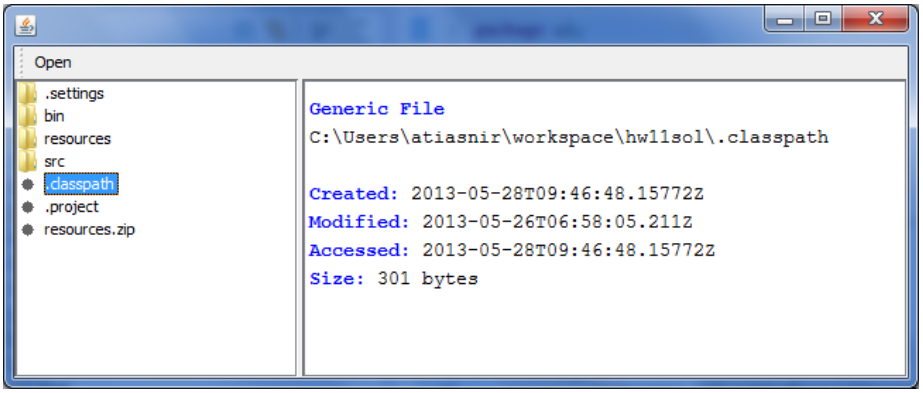
8. הסיומת tsv תסמן קבצי טקסט טבלאיים. כל שורה בטקסט מייצגת שורה בטבלה. העמודות בכל שורה מופרדות באמצעות התו טאב (\t). יש לייצר תצוגה מקדימה בצורת טבלה עבור 100 השורות הראשונות בקובץ.



9. הסיומת csv תסמן קבצי טקסט טבלאיים (בדומה ל-tsv) בהם העמודות מופרדות באמצעות התו פסיק (,). יש לייצר תצוגה מקדימה בצורת טבלה של 100 השורות הראשונות בקובץ.
10. קבצים המסתיימים בסיומת zip הם קבצי ארכיון (מכווצים) הכוללים קבצים וספריות. בתצוגה המקדימה יש לכלול את שם הקובץ וכן את שמות הקבצים המאוחסנים בקובץ הארכיון.



11. לכל קובץ אחר, כלומר כזה שהסיומת שלו אינה מזוהה ע"י המערכת, יש להציג את המידע הכללי הבא: שם הקובץ, תאריך היצירה שלו, תאריך השינוי האחרון שלו, תאריך הגישה האחרונה אליו וגודלו בבתים.



12. בכל צומת בעץ יש להקפיד ליצור את התצוגה המקדימה פעם אחת בלבד. כלומר, יש להימנע מפתיחת הקובץ מספר רב של פעמים גם כאשר המתודה renderPreview נקראת יותר מפעם אחת.
לצורך הגדרה זו, ניתן להתעלם מתצוגה מקדימה של תמונה, עבורה מעבירים את שם הקובץ המכיל את התמונה ישירות למתודה renderImage של המנשק Renderer.

<u>המנשק Renderer</u>

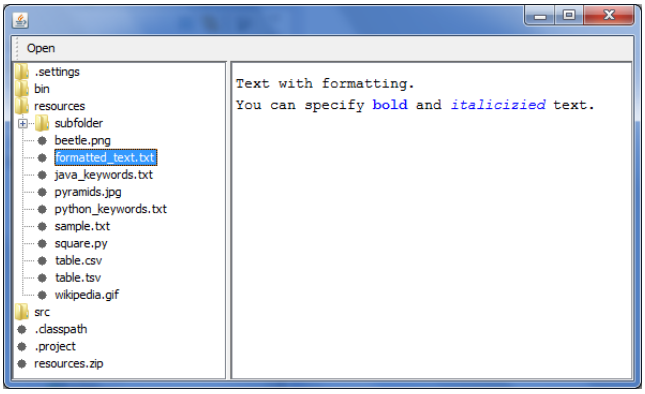
המנשק נתון ע"י ההגדרה הבאה:

```
public interface Renderer {
	public void renderText(String text);
	public void renderTable(String[][] table);
	public void renderImage(Path filename);
}
```

- המתודה renderImage מקבלת מסלול לקובץ תמונה ומציגה את התמונה.
- המתודה renderTable מקבלת טבלה בצורת מערך ומציגה את הטבלה. המימד הראשון של המערך הוא השורות, כאשר כל שורה היא מערך של עמודות.
- המתודה renderText מקבלת מחרוזת להצגה. המחרוזת יכולה להכיל הוראות מיוחדות על מנת לעצב את הטקסט:
  - טקסט שמסביבו מופיעים התגים יופיע בגופן (פונט) מודגש.
  - טקסט שמסביבו מופיעים התגים יופיע בגופן (פונט) מוטה.

  דוגמה: עבור המחרוזת "You can specify bold and <u>italicized</u> text"
  יופיע הטקסט:

  You can specify **bold** and *italicized* text



<u>הערות</u>

1. בקוד שאנו מספקים, כל המתודות במחלקה המממשת את המנשק Renderer מציגות את התוכן המתאים בחלק הימני של החלון הראשי.
2. במימוש של קריאת קובץ טקסט, אין צורך לטפל בתגים מיוחדים. לדוגמה, אם הקובץ מכיל טקסט מודגש לפי הסימנים לעיל אז התצוגה תהיה מודגשת ולא יוצג תוכן הקובץ המקורי.

חלק א' – תכנון (20 נקודות)

יש לתכנן את מימוש מבנה הנתונים באמצעות דיאגרמת מחלקות, עפ"י הדרישות שפורטו. הקפידו להימנע משכפול קוד ולהשתמש בכלים שלמדנו, כגון מחלקות אבסטרקטיות.

חשוב: אתם נדרשים לתכנן את מבנה הנתונים בלבד! כלומר מחלקות המממשות את המנשק FileSystemTreeNode. לכן, קראו תחילה (בעיון) את התיעוד של מנשק זה וכן של המנשק TreeNode, אותו הוא מרחיב.

רמז: נסו לחלוק את המימוש של הלוגיקה ליצירה חד-פעמית של התצוגה המקדימה גם כאשר סוג המשתנה הנדרש הוא שונה (לדוגמא טקסט או טבלה).

חלק ב' – מימוש המודל (70 נקודות)

יש לממש לפי התכנון בסעיף א'. כלומר, יש לממש כל טיפוס (מחלקה, מנשק וכו') שהוצג בשלב התכנון למעט אלו שסיפקנו בשלד התרגיל.

<u>הכוונה</u>:

1. יש להיעזר מחלקות Files ו-Paths אשר נמצאות בחבילה java.nio.file. בפרט:
   א. כדי למצוא את הספריות והקבצים תחת ספרייה נתונה יש להשתמש במתודה walkFileTree של המחלקה Files.
   ב. ניתן להיעזר במתודה readAttributes של המחלקה Files כדי לקבל מידע על תאריכי היצירה והשינוי של קובץ וכן את גודלו.
2. כדי לעבוד עם קבצי zip ניתן להשתמש בחבילה java.util.zip ובפרט במחלקות ZipFile ו-ZipEntry.
3. החלפה של מילים שלמות בלבד ב-java נעשית באמצעות המתודה replaceAll של המחלקה String. לשם כך יש להקיף את המילה אותה רוצים להחליף במחרוזת "\b" (בצורה זו קל יותר להתמודד עם הדגשת המילים השמורות עבור קבצי java ו-Python). נא לא להתבלבל עם התגים המדגישים שהוגדרו מקודם עבור האפליקציה שאנו בונים.

דוגמאות:

```
"integer".replaceAll("int", "INT") → INTeger
"integer".replaceAll("\\bint\\b", "INT") → integer
```

4. המתודות enumeration ו-list של המחלקה Collections עוזרות בעבודה עם המנשק Enumeration (לא להתבלבל עם enum) המוגדר ב-TreeNode.
   א. Collections.enumeration(myCollection) ממיר את האוסף myCollection ל-Enumeration.
   ב. Collections.list(myEnumeration) ממיר את myEnumeration לאוסף סטנדרטי.

חלק ג' – חיבור המודל למנשק הגרפי (10 נקודות)

הממשק הגרפי שאנו מספקים עובד עם מימושים של המנשק FileSystemTreeNode. כדי ליצור מופעים של המחלקות המתאימות, משתמשים במחלקת העזר NodeFactory (כי בזמן כתיבת ממשק המשתמש לא ידוע אילו שמות אתם בוחרים למימוש המחלקות השונות).

למחלקה זו יש מתודה בודדת, getNodeForFile, המחזירה מימוש מתאים למנשק עפ"י הקלט.

הערות:

1. ניתן לבדוק (חלקית) את המימוש בעזרת מחלקת הבדיקה TestNodeFactory הנמצאת בחבילה test.
2. לאחר מימוש המתודה ניתן לבדוק שהממשק הגרפי עובד כראוי ע"י הרצת המחלקה Application מהחבילה ui.
3. הספרייה screenshots המסופקת עם התרגיל מכילה את התמונות מהדוגמאות שלמעלה בהגדלה.

ב ה צ ל ח ה!