Rizpa Stock Exchange – README – Exercise 1

מגיש 1:

עומר פסח 318854387

[omerpessach7@gmail.com](mailto:omerpessach7@gmail.com)

מגיש 2:

ארד פאר 322593849

[arad01245@gmail.com](mailto:arad01245@gmail.com)

איך מריצים?

בקובץ הZIP נמצא קובץ הנקרא RSE Runner.bat.

הרצת קובץ זה תפתח מסך CMD דרכו נוכל לתפעל את המערכת!

תחילה, נדרש להכניס כתובת של קובץ XML אשר מכיל את בסיס הנתונים של המערכת – מידע על המניות (מספר 1 בתפריט הראשי של התוכנית).

לאחר טעינה תקינה של הקובץ, נוכל לבצע פעולות שונות בהתאם לממשק המשתמש שיוצג על המסך.

הסבר על מערכת Rizpa Stock Exchange

במסגרת קורס שפת ג'אווה ומבוא לפיתוח ווב התבקשנו ליצור מע' מסחר (דמיונית) שלמה הכוללת הן את מנוע מע' המסחר עצמו והן את הממשק המאפשר למשתמשים השונים להירשם, לנהל חשבונות, למכור ולקנות ניירות ערך וכו'.

תרגיל 1 נועד למימוש מנוע מע' המסחר המציע פונקציונלית בסיסית, תוך תפעול המע' כאפליקציית console.

חלוקה למודולים

בחרנו לממש את הפרויקט בגישה של חלוקה לשכבות לוגיות.

השכבה העסקית (Business logic - Engine). המטרה העיקרית של שכבה זו היא לבצע עיבודים לוגיים על המידע, לבצע עסקאות ואת כל הפונקציונליות של המנוע, בדיקות תקינות, קריאה מקבצים ועוד.

השכבה העליונה ביותר – שכבת הUI (User Interface). שכבה זו מקושרת אל השכבה העסקית, והיא הממשק שמוצג למשתמש. באמצעותה המשתמש יכול לבצע פעולות שונות במערכת ולעדכן את הנתונים של המערכת בסופו של דבר.

השכבה העסקית BL

שכבת המשתמש UI

בחירות המימוש שלנו

1. טעינת קובץ הXML:  
   על מנת לטעון את קובץ הXML, בחרנו להשתמש בJAXB.  
   בחירה זו נתנה לנו צורה נוחה ופשוטה לעבודה עם קובץ הXML. בנוסף בתרגילים הבאים משתנה פורמט קובץ הXML, ובחירה זו מאפשרת לנו לשנות את אופן הטיפול עם מעט מאוד עבודה.
2. Exceptions:

בחרנו לעבוד עם exceptions שנזרקים בBL מהEngine ונתפסים בUI. צורת עבודה זו מאפשרת לנו לעבוד עם Engine מודולרי, שמגיב לבקשות ופועל לפי הלוגיקה שלו, ועם מנהל UI שעובד בהתאם לEngine שאיתו הוא מתממשק.

1. קליטת מספר הפקודה מהמשתמש:

בחרנו לעבוד עם Switch – case. בחירה זו מאפשרת לנו בהמשך להוסיף פקודות נוספות בצורה קלה, לבדוק את מספר הפקודה שנבחר בצורה קלה, ולנתב לפקודה הנכונה עם הטיפול הספציפי לה בצורה פשוטה ונוחה.

1. שמירת הקובץ המכיל את מצב המערכת (בונוס):

על מנת לטעון את המערכת מקובץ שמור, על המשתמש להכניס גם את הנתיב של הקובץ אותו הוא רוצה לטעון.

בחרנו לבקש כאן גם את הנתיב (ולא לטעון מהקובץ שהמשתמש שמר לאחרונה בעזרת נתיב השמור במערכת), כיוון שרצינו לתת למשתמש אפשרות לשמור מספר קבצים עם מצב המערכת, בלי דריסה של הקובץ הקודם בכל פעם (דבר קריטי שתוך כדי מימוש עזר לנו הרבה).

בנוסף, בחרנו לשמור את הקובץ כקובץ בינארי כיוון שהמערכת היא היחידה שתעבוד עם הקובץ, ואין סיבה שיהיה קריא למשתמש/ יתפוס יותר מקום כשאין צורך .

בנוסף לכך במודול הBL קיים "מיני שכבה" שנקראת APIGateway שמטרתו היא לייצא את הפקודות הרלוונטיות, הוא המקשר "לעולם החיצון".

אופן מימוש המחלקות העיקריות

1. במודול הBL קיימים מספר packages -   
   א. apigateway בו קיימת המחלקה העיקרית APIGatewayManager – מנתבת כל בקשה למתודה האחראית עליה.
2. bl – מרכז את המחלקות שמנהלות מודלים שונים במערכת.
3. enums - מרכז את הenums השונים הקיימים במערכת (כרגע קיים אחד בלבד).
4. exceptions – מרכז את כל השגיאות שיוצרים במערכת.
5. models – מרכז את כל המודלים במערכת (חלקם מג'ונרטים אוטומטית מהJAXB – אדיר!).
6. במודול הUI קיימת מחלקת main עיקרית האחראית על ממשק המשתמש -   
   מנהלת את רצף הפעולות לפי דרישת המשתמש, מציגה תפריט ראשי למשתמש ומודיעה למשתמש על שגיאות שונות שקרו, ומנתבת את הבקשות למודל העיסקי.

בונוסים

מימשנו את שני הבונוסים האפשריים במסגרת התרגיל –

1. מימוש תמיכה בפקודת MKT – כחלק מפקודה 4 בתפריט, שבה מגישים פקודת קנייה/מכירה למע', יש למשתמש אפשרות לבחור את סוג הפקודה: LMT או MKT.
2. אפשרות למשתמש לשמור את מצב המע' לקובץ – הוספת פקודה לתפריט שתאפשר למשתמש להכניס נתיב שאליו הוא ירצה שמצב המע' יישמר – פקודה 7.

שמירת המצב של כל המניות הנוכחיות, של כל ספרי הפקודות והעסקאות שבוצעו עד עכשיו.

הוספת פקודה נוספת לתפריט שתאפשר למשתמש לטעון את כל המע' ישירות מקובץ ששמרתם אותה אליה – פקודה 8.