פרויקט JAVA – מימוש predictions כאפליקציית JavaFX

שמות המגישים:

1. שרון אולשנצקי – 318845740 – sharonol2@mta.ac.il
2. איתן פייגינה - 20653173 - eitanfe@mta.ac.il

הסבר מערכת:

* יצרנו שלושה מודולים:

**1)** **DtoManagement– ממשק ניהול הקישוריות בין הממשק הגרפי למנוע התוכנה.**

**בממשק זה יושבת מחלקת ManageSystemToUI אשר שם ביצוע בקשות המשתמש מתרחשות בהתאם לDTO שנשלח מהממשק הגרפי. בתוך המודול מחולקים שני packages שאחד מהם אחראי על קבלת הפקודות מהמשתמש ו"תרגומן" לשפת המערכת, והpackage השני אחראי "הפעלת המנוע" בהתאם לפקודת המשתמש.**

**2)System – ממשק המערכת, הממשק הראשי שמנהל את כל תהליכי המערכת.**

**3)JavaFX- ממשק שבו יושב הmain.**

**בממשק זה מתרחש כל הקשר עם המשתמש – קליטת קלט והוצאת פלט בהתאם.**

**לפי בקשת המשתמש נשלחים DTO בהתאם לSystem, דרך המקשר (DtoManagement) על מנת להניע תהליכים רצויים ולבצע בדיקות. בכל רגע נתון מוצג למשתמש רק הפקודות אשר יש בהן מהות לאותו הרגע בהתאם למצב המערכת.**

הסבר מחלקות ראשיות

* **Convert – זהו package אשר מכיל בתוכו שלל מחלקות אשר תפקידן להמיר לנו את פורמט הjaxb אל המחלקות בפורמט שלנו היושבות בsystem.**
* **ManageSystemToUI – זוהי מחלקה אשר יושבת בDtoManagement. מטרתה לנהל את הקישוריות בין ממשק המשתמש לממשק המערכת.**

**יש בה מספר מקרים (cases) שלפי ההוראה של המשתמש המחלקה פועלת בהתאם מבחינת ממשק המערכת.**

**בה בנוסף נמצאים כלל ה"עולמות" שעד כה הוטענו בהצלחה מהקובץ הנוכחי שהמשתמש העלה.**

**בנוסף נמצאים כלל ה"סימולציות" שהסתיימו בהצלחה על העולם הנוכחי – זאת על מנת לעבור בין הסימולציות השונות ולקבל מידע אודות סימולציות העבר.**

* **World – זהו package אשר מכיל בתוכו כלל המידע בנוגע לעולם.**

**בתוך ה package יש תתי תיקיות לכל אחד מסוגי המידע שיש בעולם.**

**כלומר package עבור הentities וכלל הנתונים בהם:**

**EntityDefinition – מכיל את שם הישות וכמות המופעים שלה.**

**EntityInstance – זהו מופע ספציפי של ישות בעל מזהה ייחודי. מכיל את רשימת התכונות שיש ליישות המרכזית (EntityDefinition).**

**package עבור הproperty אשר מכיל מידע עבור כל סוג של תכונה (מספר שלם, עשרוני, משתנה בוליאני ומחרוזת).**

**בתוך תיקייה זאת יצרנו interface שיודע להגריל ערכים היכן שצריך, לפי סוג המשתנה ובהתאם למגבלות השונות (כמו טווח, משתנה בוליאני, פורמט המחרוזת שהוגדר לנו).**

**package עבור הENVIORMENT אשר מכיל רשימת תכונות של משתני הסביבה.**

**package עבור הRULE אשר מכיל תתי תיקיות:**

**ACTION – שמכיל את כלל הנתונים על כל הפעולות:**

**בתוכו יש מחלקה מרכזית ManagmentExpression שתפקידה לקבל String המתאר expression כלשהו ולדלות ממנו את האובייקט המבוקש בעזרת עוד מחלקות עזר.**

**בתוך ה packageישנו interface שבתוכו יש מתודת executeAction שמהותה להפעיל כל פקודה בשעת הצורך. כל פקודה מבצעת implements לאותו interface ומממשת את מתודת executeAction בהתאם לסוגה. בכלaction יש מתודת calc שמטרתה לנהל את ניהול הישות המשניות המופעלות בכל Action בהתאם לתנאי שניתן. אותו calc קורא ל executeAction עם היישיות הראשית והמשנית בהתאם.**

**לכל סוג פקודה יש class משלו ולפקודות מאותו "סגנון" מבחינת ארגומנטים שמקבלת (כמו increase,decrease,set) יש מאין מחלקה שמנהלת אותן אשר יורשת ממחלקת הAction המרכזית.**

**בנוסף ב- package הRULE כלל הפקודות מתאכסנות כרשימה בתוך מחלקת Rule.**

**במחלקה זאת יוחזק המידע של כמות הticks עבור כל חוק וההסתברות להתרחשותו (בתור nested class).**

**בתוך הrule יש מתודה אשר מקבלת את כמות הticks בסימולציה, ומגרילה ערך בין 0 ל1 כולל. בהתאם להוראות הפעלת החוק היא יודעת להפעיל את כלל הפקודות בתוכה. אותה מתודה מבדילה בין פעולות סוף סימולציה כמו replace וkill ויודעת להעביר אותן לסט עם חשיבות לסדר על מנת שבסוף כל tick הפקודות האלו יתבצעו.**

* **מחלקת הWorld – מכילה בתוכה נתונים של כלל ה- EntityDefnition , כלל הrules ,משתני הסביבה, תנאי הסיום.**
* **Package עבור exceptions – ישנן מספר סוגי חריגות, כמו פורמט לא תקין, ערך שלא קיים, פעולה לא חוקית ועוד. בהתאם לתקלות שמתרחשות בתוכנה, היא יודעת לזרוק חריגה מתאימה עם הודעה מפורטת בנוגע לשגיאה ולסוגה.**
* **Package הsimulation – בתוך התיקייה ישנן מספר מחלקות מרכזיות:  
  1)** **SimulationRunner – אשר בתוכה מתנהל כל תהליך הרצת הסימולציה. היא מבצעת implements runnable ותהליך הסימולציה עצמה היא במתודת הrun.**

**2)** **SimulationExecutionDetails- מחלקה אשר שומרת "פורמטים" (טיפוסים) המכילים מידע שנועד להיות מוצג בתוצאות הסימולציה. מחלקה זאת ממוששת גם בתוך הWORLD כך שנתונים שמתעדכנים בWORLD מתעדכנים בהתאם גם ב SimulationExecutionDetails כך שברגע שמגיעה פקודת pull מממשק המשתמש המידע כבר מוכן עבורו.**

**3)** **SimulationExecutionManager- זוהי מחלקה שממשק קישור המערכת לממשק המשתמש (ManagmentExpression) מתחברת אליה. היא מנהלת את כל תהליך הסימולציות והתרדים. היא מכילה בתוכה מפה של מספר הסימולציה והמידע (SimulationExecutionDetails) של אותה סימולציה בהתאם.**

**4)בנוסף בתוך Package הsimulation יש עוד תת package שנקרא queue שמכיל את המחלקה: SimulationsQueueManage שהיא מכילה בתוכה הוראות ואת המידע עבור מחסנית הסימולציות.**

* **מודול JavaFx: במודול ראשית יש את המחלקה taskMain אשר היא מחלקת הmain של התרגיל. היא טוענת את הcontroller המרכזי שנקרא MainController שהוא מלווה את התוכנה לאורך כל שלביה. כל לחיצה על כפתור כלשהו (מלבד הload) תציג מיידעים אחרים על המסך בעזרת מחלקות עזר ובתוכם controllers וקבצי fxml אשר נמצאים בתוך הpackage components.**
* **אחת מהמחלקות המרכזיות בpackage זה היא מחלקת MainResultsController אשר בה נוצר תרד אשר שולח בקשות (pull) למנוע דרך הרכיב המקשר על מנת לקבל את המידע הנדרש להצגת מסך התוצאות.**
* **בנוסף יש package של pictures שמלוות אותנו לאורך כל התוכנה.**