**Predictions - תרגיל 1:**

**מגישים:** יאיר חנגלי 316590595, אופיר נקדאי 318382827

**מייל:** [yairyehezkelhe@mta.ac.il](mailto:yairyehezkelhe@mta.ac.il)

**\*מימשנו את הבונוס (שמירה וטעינה של המערכת)**

**הסבר על המערכת:**

בהפעלת התוכנית יוצג בפני המשתמש התפריט הראשי.

על המשתמש לבחור בפעולה הרצויה ע"י הקשת מספרה בתפריט.  
אפשרויות 2-3 ניתנות לביצוע רק לאחר שנטען קובץ xml תקין למערכת, או לאחר טעינת מצב מערכת שבו כבר נטען קובץ xml תקין. בכל מצב אחר תינתן למשתמש הודעה המציינת כי עליו לטעון קובץ.

לאחר בחירת אפשרות 1 - טעינת קובץ xml, על המשתמש לציין את הנתיב המלא שלו. לאחר מכן תוצג הודעה למשתמש אם הטעינה הושלמה בהצלחה ואם לא – תוצג הסיבה לכישלון.

לאחר בחירת אפשרות 2 – יוצג למשתמש מידע מסודר לגבי הסימולציה שהוגדרה בקובץ הxml.

לאחר בחירת אפשרות 3 – תוצג למשתמש רשימת משתני הסביבה והוא יוכל\*\*\*אולי לשנות בהתאם לאיך שמוצג אחרי שייעשה\*\* לבחור לספק ערך לכל אחד מהם. לאחר בחירותיו, יוצגו עבורו משתני הסיבה והערכים שנקבעו להם ותחל הרצת הסימולציה: במידה והרצת הסימולציה בוצעה בהצלחה, יוחזר למשתמש המספר הסידורי שלה וסיבת סיומה. במידה והתגלתה שגיאה בזמן הרצת הסימולציה - תינתן למשתמש הודעה המפרטת על הסיבה לכך.

לאחר בחירת אפשרות 4 – תוצג למשתמש רשימת סימולציות העבר במערכת, ממנה יבחר את זו שעליה מעוניין לקבל פרטים ולאחר מכן יוכל לבחור אילו פרטים. במידה ולא קיימות סימולציות במערכת - תינתן לו הודעה על כך.

לאחר בחירת אפשרות 5 או 6 – שמירה או טעינה של מצב קיים מקובץ, על המשתמש לציין נתיב מלא אל הקובץ, כולל שמו – ללא סיומת הקובץ. לאחר מכן תינתן הודעה למשתמש אם הטעינה הושלמה בהצלחה או לא, ואת הסיבה לכישלון.

לאחר בחירת אפשרות 7 – התוכנית תסתיים.

\*\*\*לציין גם שטעינת קובץ תקין תמחק סימולציות שכרגע קיימות במערכת או גם ככה חפרתי\*\*

**הסבר על המחלקות העיקריות ותפקידם:**

מודול SystemEngine:

המודול באחראי על תפעול המערכת מבחינה לוגית.

מחלקות עיקריות בו:

**SystemEngine** – ממשק עבור מנוע המערכת לשימוש ע"י ממשק המשתמש.

**SystemEngineImpl** – מחלקה המממשת את ממשק המנוע ומיישמת את הפונקציות השונות בו. מכילה את סימולציות העבר והנתונים הנדרשים להרצת סימולציות ויצירת DTOs עבור ממשק המשתמש בפונקציות השונות בתפריט הראש.

**WorldDefinition –** מכילה את הגדרת העולם עבור עולם שנטען מקובץ XML ושבאמצעותה נמלא את מחלקת WorldInstance בנתונים הנדרשים להרצת סימולציה באופן מעשי.

**WorldInstance –** מכילה נתוני עולם לשימוש בשלב הרצת סימולציה בהתאם להגדרת חלקיו השונים במחלקת WorldDefinition.

**EntityDefinition –** מכילה נתונים הנדרשים עבור יצירת מופעי ישות מסוימת בעולם.

**EntityInstance –** מכילה נתוני מופע ספציפי של ישות הנוצרת מהגדרתה ב EntityDefinition

**PropertyDefinition –** מכילה נתונים כלליים עבור יצירת תכונה של ישות או של משתנה סביבה בעולם.

**PropertyInstance –** מכילה נתוני מופע של תכונה ספציפית או של משתנה סביבה ספציפי. ממנה יורשות המחלקות: **BooleanProperty**, **DecimalProperty**, **FloatProperty**, **StringProperty** אשר משמשות להגדרת הערך(value) המתאים לפי סוג התכונה/משתנה הסביבה.

**Expression** – משמשת להמרת ביטוי לערך נדרש ומוודאת שערכו תואם לערך הרצוי\*\*\*\*\*(אולי לא לציין את המחלקה הזאת??)

Action – ממשק הממומש ע"י **AbstractAction**, שממנה יורשות סוגי הפעולות השונות בחוקים שבתורן מגדירות את אופן הרצתן.

**Context** – ממשק הממומש ע"י **ContextImpl** המשמש להעברת הקשר מתאים לפעולה (ישות ראשית, תכונה) לצורך הרצתה במסגרתו.

**RULES, MANAGERS, ACTIVEVARS, DIFFERENT INTERFACES...**

מודול UI:

המודול האחראי על ניהול ממשק המשתמש הקונסולי.

CLASSES OF MAIN ...... אולי אפשר להסתכל מה חן עשה ולבנות בצורה שניתן לחזור שאיתי לא יתחרפן? או מיותר כבר

מודול DTO:

בתוכו מוגדרות מחלקות DTO אשר מטרתן להעביר מידע בין שכבת המנוע לשכבת הממשק ולהיפך.

מודול Exeptions:

בתוכו מומשו חריגות שונות אותן יצרנו בהתאם לצורכי המערכת.