

תרגיל 4 - תכנות מונחה עצמים מתקדם

Design Patterns

מבוא:

בתרגיל זה אתם נדרשים להמשיך את התרגיל הקודם שמימש סוכנות רכבים. סוכנות הרכבים מחזיקה מאגר של כלי תחבורה שונים שאותם היא מעוניינת למכור ותומכת ב-GUI ובריבוי תהליכונים.

הוספת תבניות עיצוב:

עליכם לשנות את התרגיל שלכם כך שיממש לפחות את תבניות העיצוב שיפורטו בהמשך התרגיל.

שימו לב1: ייתכן שעל מנת לממש חלק מתבניות העיצוב נזכרות כאן תדרשו להשתמש בעוד כלים או תבניות עיצוב שנלמדו בכיתה.

שימו לב2: שימוש בתבניות עיצוב נוספות שלא נדרשו אך נכונות לעיצוב המערכת ותשתמשו בהן ביעילות יוביל להוספת ניקוד לציון התרגיל.

שימו לב3: תבניות עיצוב שכבר ממומשת לתוך השפה – יש להשתמש בדברים הבנויים. אין לבנות דברים קיימים בעצמכם ללא צורך.

:Abstract Factory

בעת יצירת כלי רכב חדשים לשם הכנסתם למסד הנתונים של הסוכנות, או לשם כל מטרה אחרת יש להעזר בתבנית העיצוב הנ"ל.

תוספת למחלקות:

לכל כלי רכב יש להוסיף שני נתונים נוספים:

- צבע הרכב (שישפיע על צבע המסגרת שתשורטט ב-GUI סביב אותו כלי רכב)
- סטטוס הרכב – באיזה מצב הוא נמצא כרגע: במלאי, בנסיעת מבחן, בתהליכי קניה (הסטטוס יופיע בתוצאת מתודת ה-toString)

שימו לב כי על מנת להוסיף כל אחד מאפשרויות אלו יש להיעזר בתבנית עיצוב **Decorator**.

כמו כן יש לדאוג כי ערך הסטטוס יישאר מעודכן בכל שלב – **חישוב היטב** באיזה תבנית עיצוב רצוי להשתמש פה.

נקודה למחשבה לגבי הבונוס: הוספת שדה הסטטוס המעודכן מאפשר לנו להימנע מבעיה שחלקכם נאלצתם לפתור בדיעבד בתרגיל 3. חישוב כיצד ניתן לעשות זאת.

למחלקת הסוכנות עצמה יש להוסיף "מרחק כללי", ערך זה הוא סכום כלל המרחקים שעברו כלל כלי הרכב של הסוכנות בעבר ובהווה בנסיעות מבחן. שימו לב כי על ערך זה להופיע בחלון הראשי של הסוכנות (ובו בלבד) וכי עליו להתעדכן במייד בסיום כל נסיעת מבחן בכל כלי רכב כלשהו (תבנית עיצוב **Observer**).

:Thread Pool

בשינוי מתרגיל קודם – כעת משתנה כל ניהול נסיעות המבחן. לסוכנות יש 7 מלווים שיכולים ללוות נסיעות מבחן בכל סוגי כלי הרכב. ולכן כעת הסוכנות תומכת ב-7 נסיעות מבחן במקביל, מכל סוג אפשרי, אך לא יותר מ-7 בכל רגע נתון.

כמובן שיש לממש זאת תוך שימוש בתבנית עיצוב thread pool.

:Memento

יש להוסיף כפתור נוסף ב-GUI הראשי שמאפשר שמירת מצב נתון וכן כפתור שמאפשר טעינת מצב שמור.

יש לשמור עד 3 מצבים. לאחר מכן כל שמירה חדשה גוררת מחיקה של השמירה הישנה ביותר. בכל טעינה יש לטעון את השמירה החדשה ביותר השמורה ואז למחוק אותה מרשימת מצבי השמירה.

:Singleton

המערכת שלנו תומכת ביצירת מסך ראשי אחד בלבד.

מה אתם מגישים:

מלבד הקוד המתוקן יש להגיש גם מסמך כתוב (מוקלד) שבו מצויים עבור כל תבנית עיצוב שמימשתם בקוד היכן בקוד ניתן למצוא ואתה (שם קובץ וטווח שורות) וכן הסבר על כיצד מימשתם אותה (כולל אם השתמש בדברים בנויים אז מאיזה ספריות וכו').

לתבניות עיצוב שהוספתם במימוש שלכם ולא נדרשו בתרגיל יש גם להוסיף הסבר **מדוע** לדעתכם הן נדרשות ומה הן תורמות בהוספה שלהן למערכת.

במידה ומימשתם כבר תבנית עיצוב מסויימת בתרגיל 3 – אין צורך לשנות את הקוד, אך עדיין יש צורך לציין ולעדכן את זה במסמך הכתוב.

הנחיות הגשה:

- הגשה עד לתאריך 22.06 בשעה 23:55 דרך המודל בלבד. אין הגשות באיחור!
- שאלות לגבי התרגיל יש להפנות אך ורק לתמר: tamarm@gmail.com
- במקרה של בעיה יש לפנות למרצה בלבד(!) לפחות יומים לפני הדד-ליין ולפרט מדוע מבקשים את ההארכה.
- ניתן להגיש בזוגות – רק סטודנט אחד מהזוג יגיש למודל.
- חובה לציין שמות ות"ז של כל המגישים בשני השורות הראשונות של קובץ ההגשה בהערות. כמו כן חובה ששם קובץ ה-zip של כל הקבצים יהיה מספר ת"ז (אם יותר מאחד אז מופרד על ידי קו תחתון)
- חובה לכתוב את הקוד בצורה קריאה לפי חוקי התכנות הנכון שנלמדו החל משנה א' – יורד ניקוד על חוסר ב: חלוקה נכונה לקבצים, תיעוד, שימוש מושכל בקבועים, שמות קריאים, יעילות, שבירת עקרונות התכנות המונחה עצמים, סינכרון שגוי ועוד.
- העתקה או עבודה שלא לפי חוקי וכללי המוסר יענשו בחומרה. ראו הוזהרתם.
- בהצלחה!