תכנות מונחה עצמים מתקדם

עבודת הגשה מס' 4

להגשה עד ה 06.06 ב-23:55

דגשים להגשה

- ניתן להגיש עבודה זו בזוגות – רק אחד מהסטודנטים יגיש את העבודה במודל. בתיעוד של קובץ יש לציין שם, ת.ז וקמפוס של מגיש\ים, בתוך תיעוד ה-javadoc
- לכל שאלה ניתן לפנות למרצה במייל tammarm@gmail.com . **על כל פניה** להכיל את פרטי הסטודנט המלאים כולל שם מלא ות.ז.
- חובה לתעד כל קובץ, מחלקה ופונקציה ע"י javaDoc. ניתן להיעזר בתיעוד באתר oracle או בקבצים הרלוונטיים במודל.
- העבודה מתבססת על עבודת הגשה 3 – עליכם לעדכן/להרחיב את המחלקות הקיימות במידת הצורך ולהשתמש בהן. כל הדרישות מעבודה 3 רלוונטיות לעבודה זאת אלא עם נאמר אחרת.
- בכל מקרה שאין פירוט מדויק מה לעשות הכוונה היא שיש לכם חופש בחירה, אל תשאלו שאלות קיטבג'! אך שימו לב שחובה לבחור בפתרון השומר על חוקי התכנון הנכון וגם על עקרונות התכנות המונחה עצמים. פתרונות שיפרו עקרונות אלו יחשבו שגויים!
- בכל מקום שלא נאמר במפורש אחרת על השינויים להיות מבניים בלבד, זאת אומרת שהם אינם אמורים להשפיע על הקלט/ פלט/ ביצוע של המערכת.

1. מבוא

זהו התרגיל הרביעי והאחרון בקורס, בתרגיל זה תתרגלו תבניות עיצוב (Design patterns).

חשוב להבין: תרגיל זה לא יכיל מימוש של חלקי עבודה חדשים מעבר לתרגיל 3, אלא יהיה תרגיל של הבניית (re-constructing) הקוד מחדש תוך שימוש בתבניות עיצוב.

ומכך נובע שלמעט במקומות שדבר זה ייכתב במפורש, אין שינוי בקלט, פלט או צורת העבודה של הקוד מתרגיל 3.

2. חלק א- הבניית הקוד

עליכם לשנות את מימוש תרגיל 3 כך שיתמוך בתבניות העיצוב הבאות:

א. <u>Thread Pool</u>

בתרגיל 3 כל חיה הייתה Runnable, והחזיקה בתוכה Thread שישמש כ- executer עבורה.
כמו כן ה-GUI תמך ביצירה של עד 10 חיות ומעבר לזה נתן התרעה שלא ניתן לייצר יותר.
בתרגיל 4 יש לשנות זאת שכל חיה תהיה Runnable, אך לא תחזיק בתוכה thread. במקום ה-GUI יחזיק אובייקט Thread pool – שיבנה על ידיכם בצורה הבאה:

הוא יכיל 10 Threads שתפקידם להריץ את החיות. כשחיה מתה (מכל אחת מהסיבות שנמנו בתרגילים הקודמים) ה-Thread שהריץ אותה מתפנה והולך לחפש חיה אחרת להריץ מהתור. כל ניסיון לייצר חיה מעבר ל-10 שכבר רצות מכניס את החיה החדשה לתור של ה-Thread Pool.

שימו לב: התור של ה-Thread Pool יהיה בגודל מקסימלי של 5 חיות.
כמו כן בסיום המערכת וכיבוי ה-GUI ה-thread-ים יכובו וה-Thread pool תיסגר בצורה מסודרת.

ב. Singleton

גם מחלקת ZooPanel וגם המאכלים השונים (אלו שמומשו כאובייקטים בתרגיל 3) יכולים להופיע רק פעם אחת בזמן נתון במערכת. ולכן יש לממש מחלקות אלו כ-singleton.

ג. Abstract Factory

בעת בקשה לייצר חיה על ידי ה-GUI, יש לשנות שקודם המשתמש יבחר את סוג החיה מבין ב-**Carnivore** או **Omnivore** או **Herbivore** (בחירת factory). לאחר הבחירה הזאת יוכל המשתמש לבחור את החיה הספציפית ולספק את שאר הפרטים.
שימו לב כי כל התהליך היצירה של החיות יבוצע תוך שימוש בתבנית העיצוב abstract factory.

ד. Decorator

יש לממש את תבנית העיצוב decorator על חיה כך שנוכל לתמוך בשינוי צבע החיה (שינוי הצבע ימומש על ידי שינוי ה-image). שימו לב שיש גם להוסיף אפשרות ב-GUI שתאפשר לנו לשנות את הצבע של חיה ספציפית.

ה. Observer / Listener

בתרגיל 3 התבקשתם לממש Controller thread כ-data member של מחלקת ZooPanel.
בתרגיל 4 אתם מתבקשים לשנות את המימוש כך שה-controller יהיה מחלקה עצמאית שמממשת את הממשק observer.
כל חיה תהיה מסוג observable, וה-controller יאזין לכל החיות ויקבל מכל אחת מהן דיווחים על כל שינוי הדורש זאת. ובהתאם יבצע את הנדרש לפי ההנחיות בתרגיל 3.

ו. Memento

יש להוסיף ב- GUI אפשרויות של "שמירת מצב" ו-"שחזור מצב". אפשרויות אלו ימומשו תוך שימוש בתבנית העיצוב memento.
המימוש שלכם יאפשר שמירה של עד 3 מצבים קודמים.
לאחר "שחזור מצב" – הוא נמחק מהמצבים השמורים, ובקשת השחזור הבאה תשחזר את המצב שנשמר לפניו.
במידה ויש בקשה ל"שחזור מצב" בלי שהייתה קודם בקשה ל"שמירת מצב" – יש לתת הודעת שגיאה מתאימה.

3. חלק ב – מסמך pdf

בנוסף לקוד המתוקן יש להגיש מסמך pdf. במסמך יופיעו הפירוטים הבאים:

1. הסבר עבור כל תבנית שהוזכרה בחלק א - היכן בדיוק בקוד שיניתם על מנת לתמוך בתבנית.

2. הסברים על תבניות עיצוב נוספות שיש לממש (או כבר ממומשות מאחורי הקלעים) עבור המערכת. לכל תבנית עיצוב יש להסביר היכן יש לממשה (או היכן היא כבר ממומשת) ומדוע. נקודות בונוס יינתנו על מימוש ממש של תבניות נוספות בקוד בחלק א.

3. יש לפרט לפחות 3 תבניות עיצוב שלא מתאים לממשם במערכת. יש לפרט עבור כל אחת שציינתם מדוע היא אינה מתאימה בשום פנים למערכת ואף סותרת את הלוגיקה שלה. אנחנו מצפים לראות מחשבה יצירתית והנמקות לוגיות במענה לשאלה זאת!

בהצלחה!