

# תכנות מונחה עצמים מתקדם

## עבודת הגשה מס' 4

להגשה עד ה 07.06 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 אנא עברו על מסמך ה-FAQ ובדקו האם היא נענתה כבר. במידה ולא, ניתן להוסיף הצעת עריכה בקובץ או לפנות למרצות האחריות במייל [marinal@ac.sce.ac.il](mailto:marinal@ac.sce.ac.il) או למתרגל סרגיי טרגולוב (בשאלות הקשורות למימוש) במייל [tammarm@gmail.com](mailto:tammarm@gmail.com) [sergeit@ac.sce.ac.il](mailto:sergeit@ac.sce.ac.il)
- על כל פניה להכיל את פרטי הסטודנט המלאים כולל ת.ז., קמפוס, ושם המתרגל.
- 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.

### 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.

ניתן להשתמש בטיפוסים מובנים של Java כמו: Observer, Observable, ThreadPool, Executors, etc. יש לעיין בחבילה [java.util.concurrent](http://java.util.concurrent)

### 2. הבניית הקוד

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

#### א. Thread Pool

בתרגיל 3 כל חיה הייתה Runnable, והחזיקה בתוכה Thread שימש 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.

עליכם להוסיף כפתור בשם "Clear" ל-ZooPanel. לאחר לחיצה על הכפתור, עליכם לגרום לכל התהליכונים להפסיק להריץ את החיה הנוכחית (ללא אפשרות לחזור אליה) ולעבור לחיות הממתינות בתור (אם יש). אם לא נמצאות חיות ממתינות בתור, התהליכונים של ה-Thread Pool ימתינו לחיות החדשות.

## **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.

## **Abstract Factory**

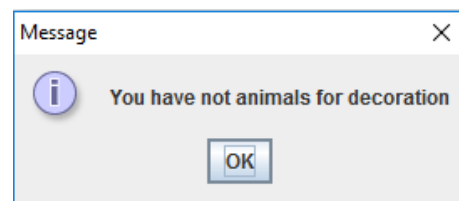
בעת בקשה לייצר חיה על ידי ה-GUI, יש לשנות שקודם המשתמש יבחר האם מדובר ב-Carnivore או Omnivore או Herbivore. (בחירת factory) לאחר הבחירה הזאת יוכל המשתמש לבחור את החיה הספציפית ולספק את שאר הפרטים. שימו לב כי כל התהליך היצירה של החיות יבוצע תוך שימוש בתבנית העיצוב abstract factory.

מחלקות CarnivoreFactory, OmnivoreFactory, HerbivoreFactory צריכות לממש את אותו ממשק AbstractZooFactory בעל מתודה produceAnimal(type). עליכם להגדיר מתודה createAnimalFactory(foodtype) שתחזיר את המחלקה של הפקטורי המתאימה (מבין CarnivoreFactory, OmnivoreFactory, HerbivoreFactory).

## **ג. Decorator**

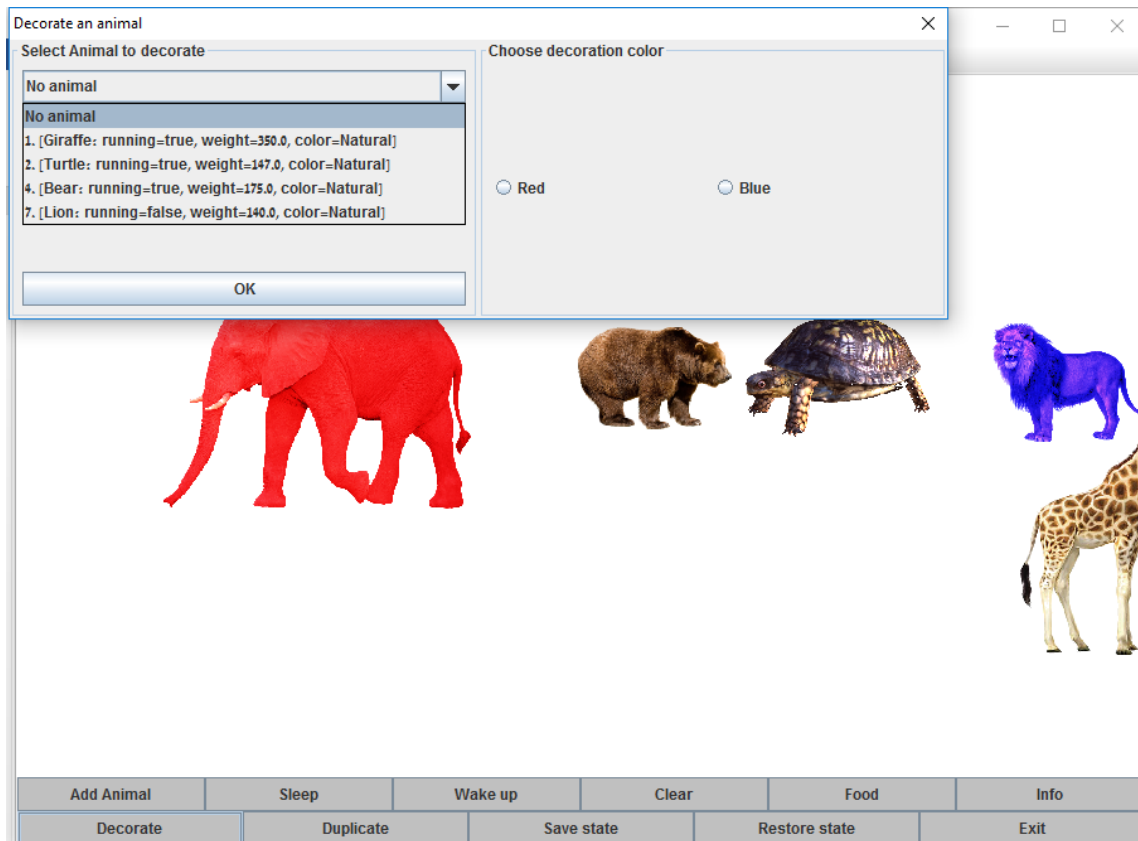
יש לממש את תבנית העיצוב decorator על חיה כך שנוכל לתמוך בשינוי צבע החיה (שינוי הצבע ימומש על ידי שינוי ה-image). שימו לב שיש גם להוסיף אפשרות ב-GUI שתאפשר לנו לשנות את הצבע של חיה ספציפית רק במקרה שצבע של חיה הוא טבעי. עליכם להוסיף כפתור בשם "Decorate" ל-ZooPanel. לאחר לחיצה על הכפתור, עליכם:

1. אם אוסף של חיות רק או באוסף אין אף חיה בצבע טבעי אז להודיע על זה:



2. אם יש חיות בצבע טבעי, אז ליצור Dialog חדש בשם "DecorateDialog" ובו ב-JComboBox נתונים של כל החיות שבצבע טבעי (מצב של תהליכון של חיה ב-ThreadPool, משקל וצבע).

3. על הפאנל שיצרתם עם רשימת החיות יש להוסיף שני כפתורים לשינוי צבע לכל חיה: "Red" ו-"Blue". יש לבחור צבע.



4. לאחר לחיצה על הכפתור "OK" יש להפעיל את פונקציה של Decorator DP שמשנה צבע של חיה שנבחרה מ-JComboBox.

המטרה יצירת מחלקה האחראית על הוספת צבע לחיה.

יש ליצור את המחלקות/הממשקים הבאים:

- ממשק בשם ColoredAnimal המכיל פונק. PaintAnimal()
  - מחלקה בשם ColoredAnimalDecorator מממשת את הממשק ColoredAnimal ומכיל משתנה מהסוג הנ"ל (ColoredAnimal). מחלקה זו תממש את הפונקציה paintAnimal() בכך שתוסיף כל פעם צבע חדש לחיות.
- מחלקה Animal מממשת את ColoredAnimal .

## 7. Observer / Listener

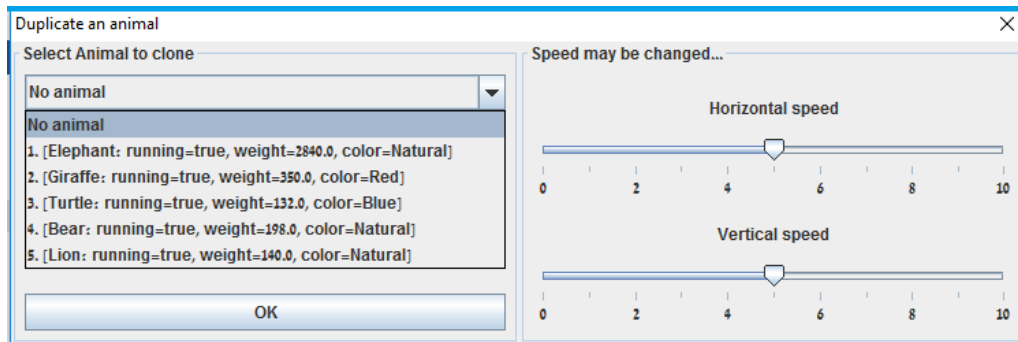
בתרגיל 3 התבקשתם לממש Controller thread כ-data member של מחלקת ZooPanel. בתרגיל 4 אתם מתבקשים לשנות את המימוש כך שה-controller יהיה מחלקה עצמאית בשם ZooObserver שמממשת את הממשק [Observer](#). כל חיה תהיה מסוג [Observable](#), וה-controller יאזין לכל החיות ויקבל מכל אחת מהן דיווחים על כל שינוי הדורש זאת.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

## ה. Prototype

עליכם לממש תבנית עיצוב זו, ע"י שכפול המופעים של Animals באופן הבא:

- יש להוסיף על ה ZooPanel - כפתור בשם "Duplicate"
- לאחר לחיצה על הכפתור, יש לפתוח חלון בשם "DuplicateDialog" ובו תישאלו איזה חיה לשכפל (ב- JComboBox) וגם אפשרויות ל עדכון של התכונות הבאות:

- מהירות אופקית (פיקסלים ביחידות זמן) מ-0 עד 10
- מהירות אנכית (פיקסלים ביחידות זמן) מ-0 עד 10

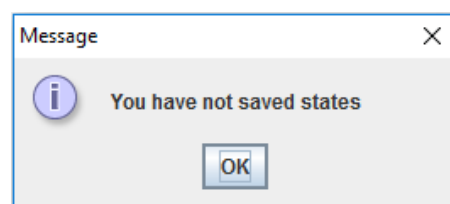


## ו. Memento

יש להוסיף ב- GUI אפשרויות של "שמירת מצב" ו-"שחזור מצב" (יש להוסיף ל- ZooPanel שני כפתורים "Save state" ו- "Restore state"). 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. יש ליצור מחלקה בשם ZooMemento לשמירת אוסף של חיות ואוכל (אם יש). במידה ויש בקשה ל"שחזור מצב" בלי שהייתה קודם בקשה ל"שמירת מצב" – יש לתת הודעת שגיאה מתאימה:



### נקודות חשובות בנוגע לכלל המימושים:

- 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.
- ניתן להיפטר (ולא להגיש) מחלקות/שדות/מתודות שלא חיוניים למימוש העבודה הזאת.

בכל שאלה בנוגע לתרגיל יש לפנות לאחראים התרגיל:

מרינה ליטבק [marinal@sce.ac.il](mailto:marinal@sce.ac.il)

תמר שרוט [tammarm@gmail.com](mailto:tammarm@gmail.com)

סרגיי טרגולוב (בשאלות הקשורות למימוש) [sergeit@ac.sce.ac.il](mailto:sergeit@ac.sce.ac.il)

## בהצלחה!