תכן ותכנות מונחה עצמים – 046271

תרגיל בית 4

<u>מגישים</u>:

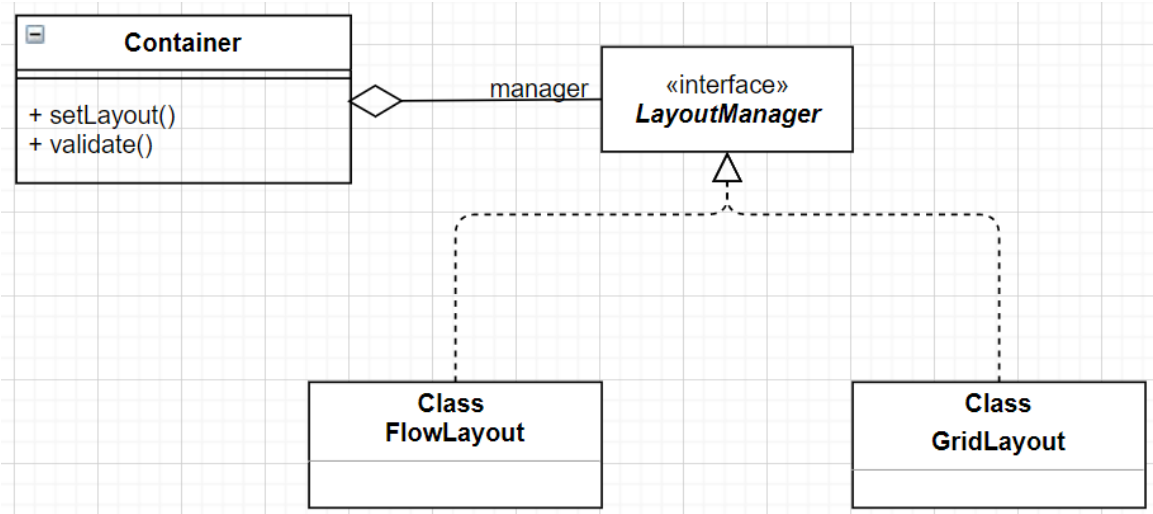
שם	תעודת זהות
בנימין סרוסי	311314975
יואב כהן	203115373

שאלה 1

סעיף א' –

ה design pattern שממומש בשאלה הוא מסוג Strategy. הסיבה לכך היא שהבעיה המוצגת היא החלפת ה- Layout על ידי המשתמש בזמן הריצה בצורה גמישה.
נשים לב כי יש לנו 2 מחלקות אשר מממשות את אותו הממשק והן קשורות אחת לשניה ונבדלות בהתנהגות שלהן.

סעיף ב' –



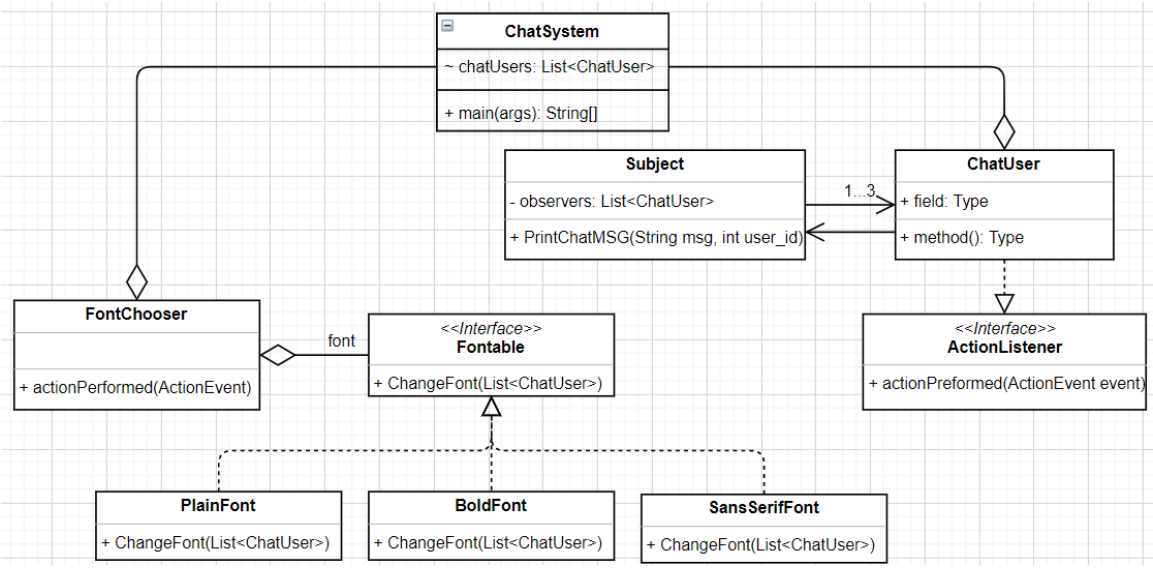
שאלה 2

הסבר על התכן והמימוש:

לצורך המימוש הגדרנו מספר מחלקות וממשקים המגדירים את ממשק המשתמש:

1. Chat user : מחלקה זו מדמה את המשתמש בשיחה . כל משתמש מיוצג בתוך JPanel . ה- JPanel מורכב משני Jlable שמייצגים תיבת טקסט ותיבת צ'אט שבה ניתן לראות את השיחה. עיקרון התכן שבו השתמשנו הוא Observer בו שדה הטקסט, עוקב אחרי המשתמש . ברגע שהמשתמש מקיש על המקש Enter מתבצעת המתודה actionPreformed ואז הסובייקט מציג את ההודעה שהמשתמש הקליד על המסכים של כל המשתמשים.

2. Subject : מחלקה שמטרתה לממש את הסובייקט בעיקרון התכן שבחרנו . המחלקה מכילה רשימה של כל משתמשי המערכת ויודעת לעדכן אותם בהתאם ולהציג את הטקסט.
3. ממשק Fontable ממשק המכיל בתוכו מתודה אחת changeFont . את הממשק ממשות שלוש מחלקות SansSerifFont BoldFont PlainFont אשר תפקידן לשנות את הגופן לגופן הרצוי ברגע לחיצה על הכפתורים . אנו מחזיקים מופע של הממשק במחלקה FontChooser וכך ממשים את עיקרון התכן strategy.
4. מחלקת FontChooser מחלקה הממשת את כפתורי המערכת בקשר לשינוי הגופן.
5. מחלקת ChatSystem המחלקה הראשית שמכילה את המופעים של היוזרים ושם מגדירים את ה-Jframe . בנוסף זו המחלקה בה אנו מבצעים טסטים לבדיקת המערכת שלנו

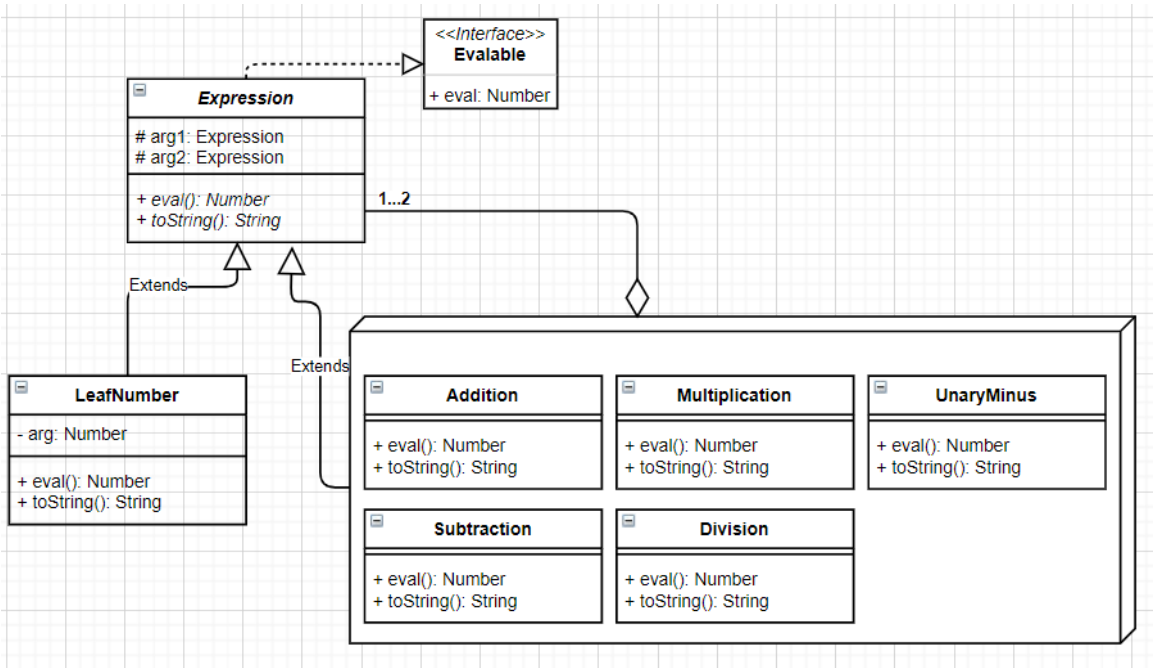


שאלה 3

ה design pattern שבחרנו הוא מסוג Composite.
הסיבה שבחרנו בו היא זיהוי של מבנה מסוג עץ בבעיה הכולל קריאה לפונקציאנליות זהה עבור העלים ועבור הצמתי שבעץ. כלומר עלנו להתייחס לאובייקטים פשוט ומורכבים בצורה זהה.

בתרגיל שלנו אובייקט פשוט הוא מספר (מסוג שלם או שבר) ואובייקט מורכב הוא מסוג טיפוס אריתמטי (חיבור, חיסור, כפל, חילוק ופעולת המינוס האונארי).

טיפוסים מורכבים בנויים מעוד ביטויים מורכבים או מטיפוסים פשוטים של מספרים.



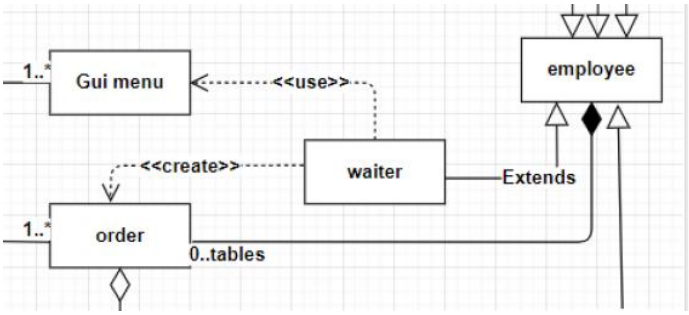
שאלה 4

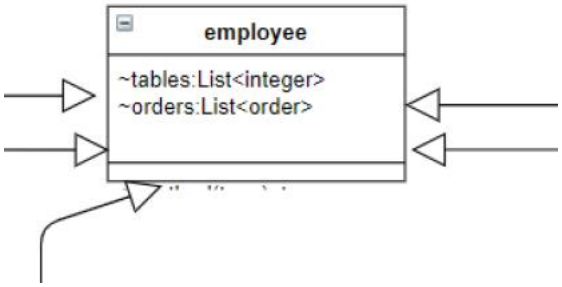
בתרגיל בית 3 הגדרנו מחלקת employee אשר ממנה יורשים כלל משתמשי המערכת במסעדה.
מחלקה זו מחזיקה את כל המידע על ההזמנות הקיימות במערכת ומצבן, המידע על כלל השולחנות במסעדה ומצבם.

מסיבה זו מחלקת employee מקיימת את עיקרון ה Information Expert בתכן של התוכנה שעשינו.

מחלקת waiter שיורשת ממחלקת employee היא זאת אשר יוצרת הזמנות חדשות, מטפלת בסטטאוס של ההזמנות ומעדכנת אותן.
מסיבה זו מחלקת waiter מקיימת את עיקרון ה creator בתכן של התוכנה שלנו.

ניתן לראות זאת מהתרשימים הבאים:

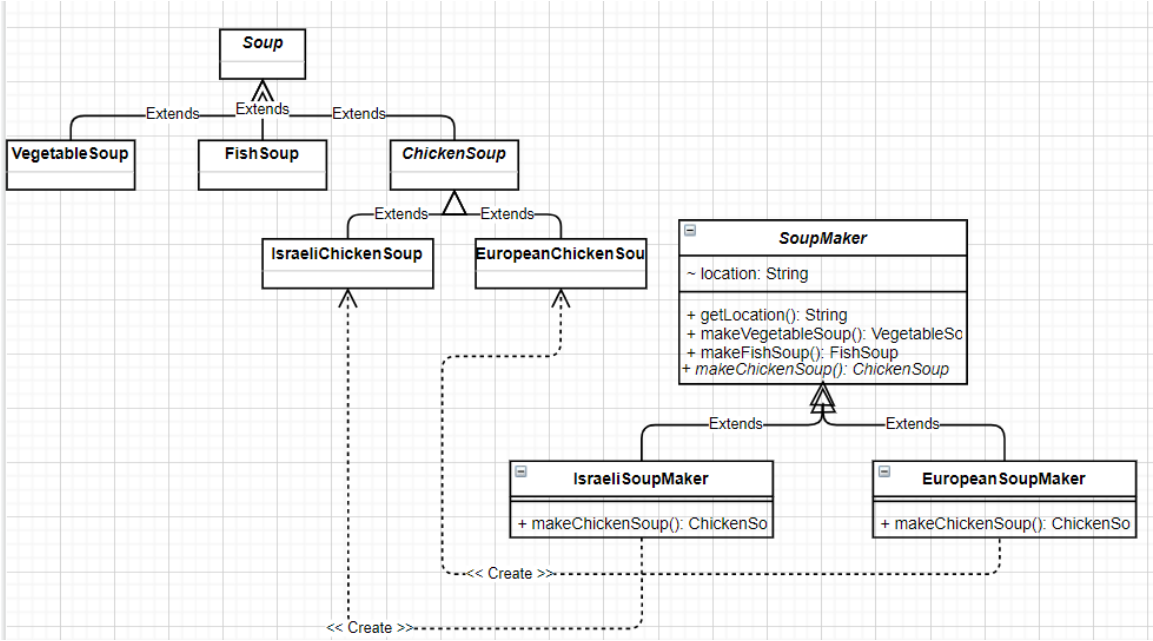




שאלה 5

סעיף א' –

ה design pattern שמממוש כאן הוא factory. בעיית התכן שאנו פותרים בעזרת design pattern זה הוא הפרדת יצירת האובייקטים השונים שלנו והקונפיגורציה שלהם. בעזרת מימוש זה אנחנו מורידים את רמת הצימוד של התכן שלנו, ומאפשרים רמת אבסטרקציה של יצירת אובייקטים.



סעיף ב' –

מחלקת Calendar מממשת את ה design pattern מסוג singleton.

ניתן לראות כי הקריאה הראשונה היא למתודה של `getInstance` , זהו מימוש של `singleton` כמו שלמדנו בהרצאה. מימוש זה הגיוני עבור אובייקט מסוג לוח שנה אשר כדאי שיהיה מופע בודד שלו שכל רכיבי התכנית יוכלו להשתמש בו.