# תכנות ותכן מונחה עצמים
# 046271

## דו"ח הגשה – תרגיל בית רטוב 4


מתן צחור 208936989

פז וולף 206555138

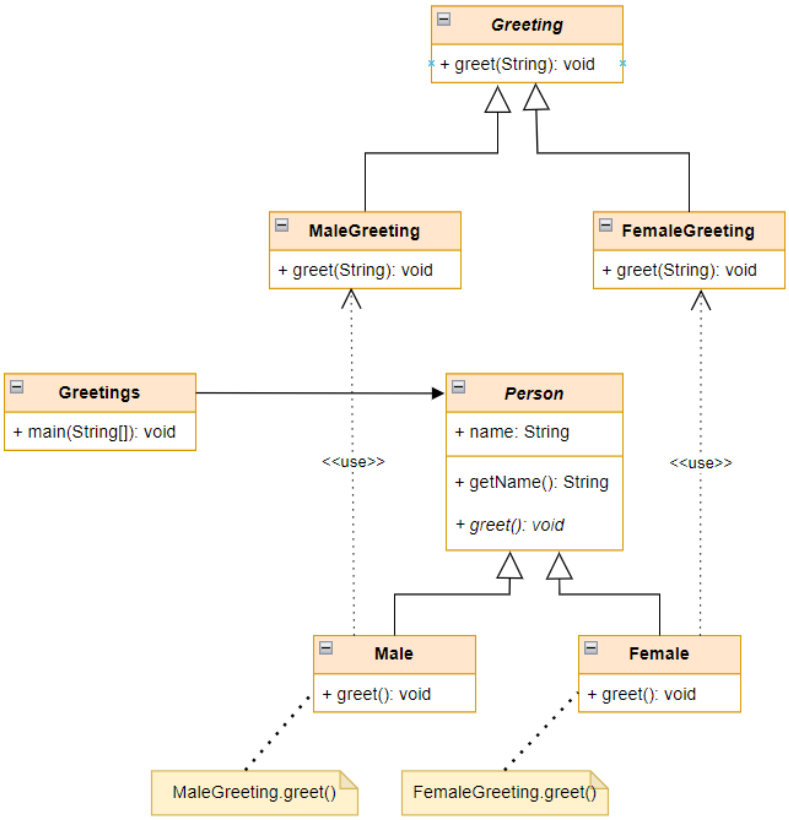
# שאלה 1

א. התוכנית הנתונה תדפיס:

Hello Mr. Danny
Hello Ms. Danna

ב. ה-$design\ pattern$ בו נעשה שימוש בקוד הנתון הוא $Decorator$.
ה-$design\ pattern$ הזה בא להוסיף פונקציונליות לקוד קיים תוך כדי שמירה על הממשק המוכר לאותו קוד. בקוד הנתון ניתן לראות זאת בכך שכל היורשים מהמחלקה $Person$ מממשים ממשק אחיד לשימוש במתודה $greet()$ באופן שעושה שימוש במחלקות העוטפות של $Greeting$. כל מחלקה שעוטפת את $Greeting$ מממשת את $greet()$ בצורה שעושה שימוש במימוש המקורי שלה ומוסיפה עלייה פונקציונליות נוספות – הדפסת הקדמה לשם המודפס.

ג. להלן מצורף $class\ diagram$ עבור המחלקות הנתונות:




ניתן לראות מתוך התרשים את אופן השימוש ב-$Decorator$ בכך שהמחלקות היורשות מ-Person משתמשות בגרסאות "עשירות" יותר של $Greeting$, ע"י שימוש במחלקות העוטפות שלה - $MaleGreeting$ ו-$FemaleGreeting$.

## שאלה 2

בקוד שלנו באים לידי ביטוי ה-$design\ patterns$ הבאים: $Singleton, Observer, Strategy$ באופנים הבאים:

$Singleton$ – המחלקה $ColorGenerator$ למעשה ממומשת כ-$Singleton$. ייתכן מופע יחיד שלה בכל התוכנית בכך שהבנאי שלה הוא $private$, והדרך היחידה לפנות לאובייקט מהמחלקה היא דרך קריאה למתודה $getInstance()$ שמחזירה את המופע היחיד של המחלקה השמור כשדה סטטי של המחלקה (או יוצרת אותו במקרה וזו הקריאה הראשונה למתודה בתוכנית).

$Observer$ – המחלקה $ColorGenerator$ מממשת את ה-$design\ pattern$ הזה בכך שהיא מחזיקה תחתיה רשימה של "מאזינים". בכל פעם שהמחלקה רוצה לעדכן את צבעה, היא מחליפה אותו ואז מודיעה לכל רשימת ה"מאזינים" שלה שהצבע עודכן, ע"י קראיה למתודה $updateColor(Color)$ שכל "מאזין" מממש. מתודה זו היא חלק מה-$interface$ שמימשנו - $ColorObserver$. כך כל "מאזין" יודע להתעדכן בהתאם לקבלת ההודעה מהמחלקה $ColorGenerator$.
תכן זה מאפשר לנו לקבל עדכונים שבצורה אסינכרונית מה-$ColorGenerator$ לגבי מתי הזמן המתאים לעדכן את צבע הלוח.

$Strategy$ – ה-$interface$ שכתבנו $ColoringStrategy$ עוזר לממש תכן זה בכך שכל מחלקה שמממשת אותו יוצרת אופן אחר לסדר עדכון ה-$Panles$. המחלקה $ColorGenerator$ מחזיקה מופע של אובייקט $strategy$ המממש $ColoringStrategy$ ולפיו קובעת מה הסדר שבו היא שולחת עדכונים ל"מאזינים" שלה. האובייקט $strategy$ קובע את הסדר שבו מתקבלים האינדקסים של ה-$Panels$ הבאים לעדכון, והוא ניתן להחלפה בכל רגע נתון. כלומר, אלגוריתם קביעת הסדר נקבע לפי אובייקט זה.
תכן זה מאפשר לנו לקבוע את סדר העדכון בצורה דינאמית שניתנת לשינוי ועדכון ולא מצריך מאיתנו לכתוב אובייקט $ColorGenerator$ שונה לכל שיטת סידור שונה.