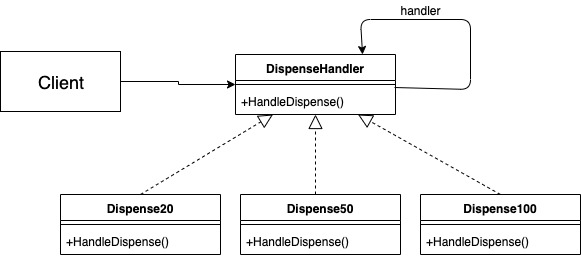
**עבודה 2 – יסודות הנדסת תוכנה**

שאלה 2

א. נניח שאנחנו צריכים לממש את מנגון משיכת הכסף בכספומט.

המשתמש מזין כמה כסף הוא רוצה להוציא, והכסף מוחזר בשטרות של 100,50,20. נניח שיש לנו מתקן עבור שטרות של 100, מתקן עבור שטרות של 50 ומתקן עבור שטרות של 20, וכמובן נניח שאנחנו רוצים להחזיר שטרות לפי הסדר הבא : 100 , 50 ו 20.

לכן שרשרת האחריות שלנו תהיה : החזר שטרות של 100 -> החזר שטרות של 50 -> החזר שטרות של 20, כך שכל חולייה בשרשרת תנסה לקחת כמה שיותר שטרות מהסוג שלה, ואם היא לא מצליחה היא **תעביר את האחריות לחולייה הבאה**.(נניח כי סכום הכסף המבוקש הוא תמיד כפולה של 10). כמובן שבמימוש קונקרטי כל חוליה כזאת תממש את הממשק Handler והפונקציית handle() שלה היא זו שתבצע את הלוגיקה שמפורטת הנ״ל.



ב.

תפקיד ה Handler - DispenseHandler שמהווה את הממשק אותו ממשים שאר מנפקי הכסף. המתודה HandleDispense() מחליטה כמה שטרות לקחת מאותו סוג ואז קוראת למנפק הבא.

תפקיד ה ConcreteHandler – הם ה Dispense20, Dispense50, Dispense100 שממשים את הממשק.

תפקיד ה Client – המשתמש אשר רוצה למשוך כסף ומזין את הסכום המבוקש.

ג. דמיון :

* בשתי המימושים אנו מקבלים מבנה דומה, של מעין רשימה מקושרת, כאשר כל אובייקט מכיל את האובייקט הבא בשרשרת שלו יש לקרוא.
* שני תבניות העיצוב בוחרות בקומפוזיציה כפתרון **במקום הורשה** כדי לספק פתרון שהוא יותר גמיש בזמן ריצה.

שוני :

* ב Chain Of Responsibility אנו עוצרים את הקריאות ל successor ברגע שהגענו לחולייה אשר יכולה להתמודד עם הקלט, לעומת זאת ב Decorator אנו ממשיכים את הקריאות עד להגעת למחלקת האב.
* ב Chain Of Responsibility אנו משתמשים כאשר אנחנו רוצים להתמודד עם הודעות שונות, ואילו ב Decorator אנו משתמשים כאשר אנו רוצים להוסיף **פונקציונליות או מידע חדש**.

ד. מימוש של תוכנה לחברה המכינת פיצות, כך שיש סוגי פיצות שונות וסוגי תוספות שונות.

נשתמש בתבנית העיצוב Decorator על מנת להוסיף או להסיר פונקצינאליות שונה לכל פיצה, נעשה זאת כך שתהיה לנו פיצה בסיסית (לדוגמא רק עם גבינה) ועליה נוסיף עוד סוגים/תוספות.

דוגמא זאת לא מתאימה ל Chain Of Responsibility מכיוון שכל חוליה בשרשרת היא קריטית (מהווה תוספת לפיצה) ומוסיפה מידע חדש וחשוב, לעומת זאת Chain Of Responsibility נועדה על מנת להתמודד עם הודעות וברגע שהגענו לחוליה בשרשרת אשר יודעת להתמודד (handle) עם ההודעה, היא מטפלת בה ועוצרת את התהליך.