

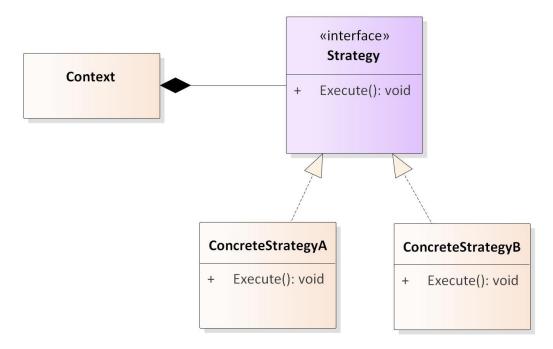
מאפייני המערכת:

- החניונים המנוהלים יכולים להיות רב קומתיים
- כל קומה יכולה להכיל סוגי חניה שונים (לאופנועים, קומפקטי, גדול...) כך שלכל סוג רכב יש מקומות חניה מוגדרים.
- בעת ביצוע החניה בעל הרכב מקבל כרטיס חניה, ומאוחר יותר כאשר ירצה לשלם יוכל לעשות זאת הן במזומן והן
 באשראי.
 - ישנה גמישות מבחינת בחירת ושינוי הנוסחה לחישוב העלות ע"י אדמינים.
 - במערכת שלנו אדמינים יוכלו להוסיף/להסיר קומות ומקומות חניה ע"פ הצורך.
 - לוח דיגיטלי יוצג בכניסה לחניון ויפרט את הסטטוס בקומות השונות.
 - המערכת שלנו היא זו שתקבע עבור כל רכב בתוך הקומה הרצויה את המיקום המדויק של החניה, כאשראסטרטגיית השמה זו תהיה גמישה ותוכל להיות מוחלפת באחרת בכל שלב.

Design Patterns

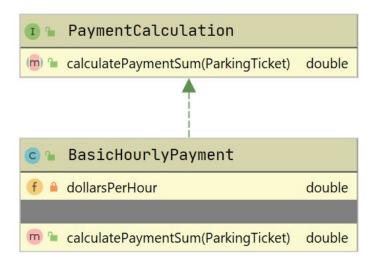
Strategy Pattern

תבנית המאפשרת בחירת האלגוריתם הרצוי בזמן ריצה. על מנת לעשות זאת מגדירים אלגוריתמים
 כמחלקות המממשות ממשק אחיד ושולחים את האלגוריתם הרצוי כפרמטר.

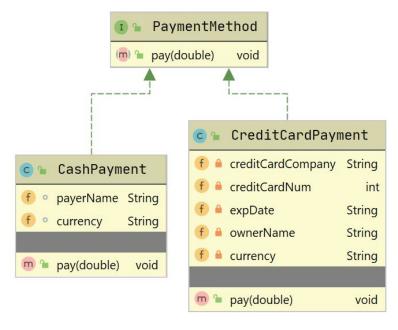


Strategy Pattern

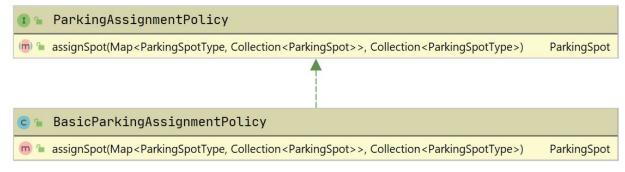
ניתן לשנות את המדיניות לחישוב הסכום לתשלום בכל זמן שנו

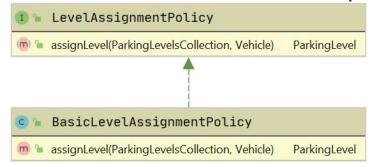


■ בעל הרכב יכול לשלם הן במזומן והן



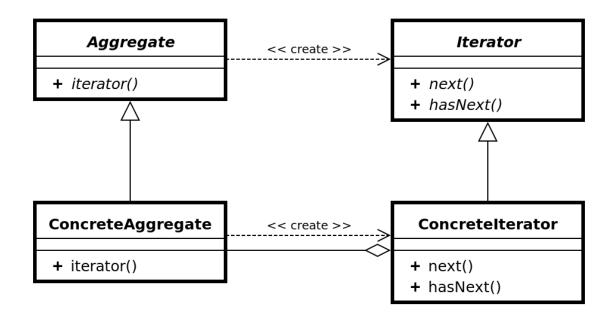
● ניתן לשנות את המדיניות של בחירת הקומה לחנייה והמדיניות של בחירת מקום החנייה עצמו בתוך הקומה בכל זמן שנרצה



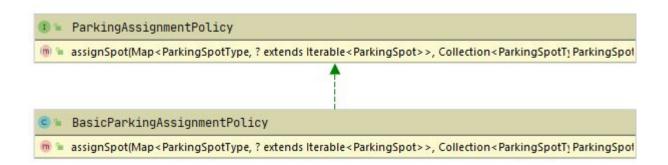


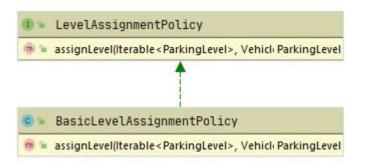
Iterator Pattern

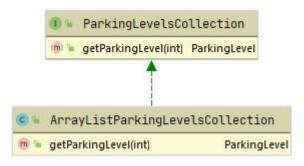
 ממשק המאפשר מעבר נוח על מבני נתונים ללא צורך להכיר את המימוש הפנימי (והאולי מורכב) של המעבר הזה.



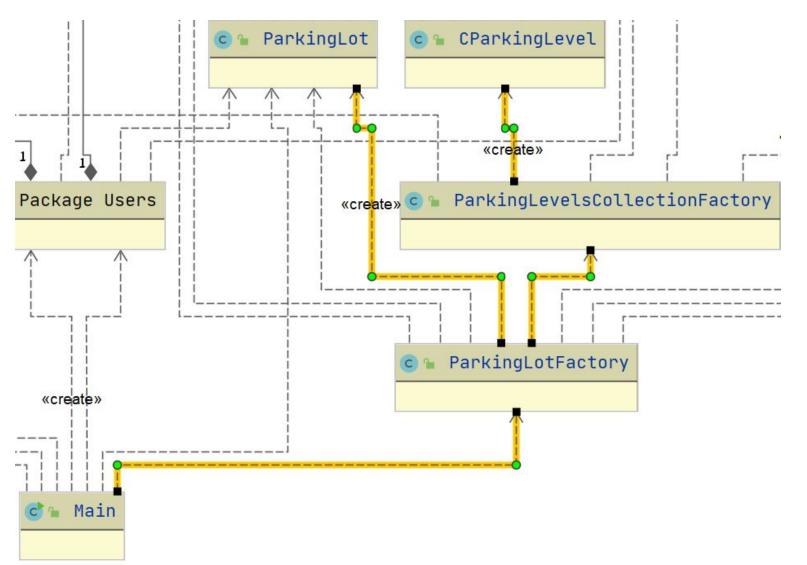
Iterator Pattern







Static Factory Method Pattern

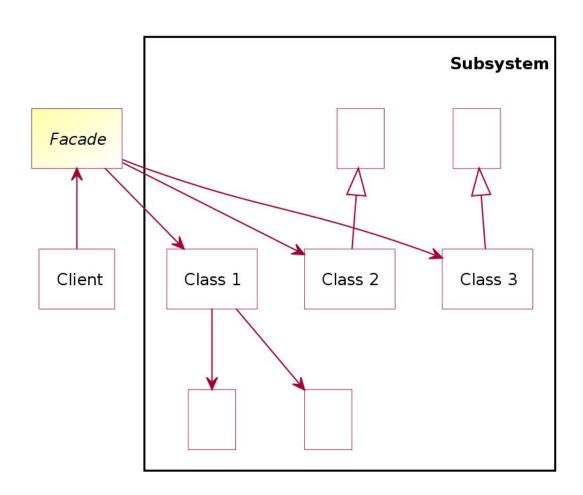


● החוא לא צריך להסתבך עם יצירה של אובייקטים מורכבים. הוא קורא למתודה בParkingLotFactory שיוצרת עבורו את ParkingLot.

ParkingLotFactory

factory בתורו, משתמש בתורו, משתמש אחר שמייצר אוסף של קומות בחניון.

Facade Pattern



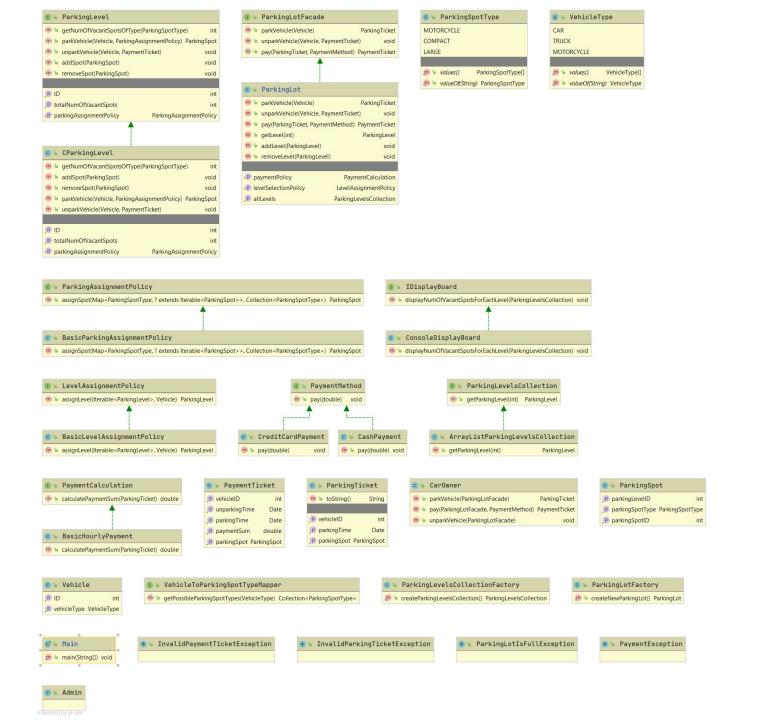
מאפשר ממשק יחיד ומרוכז שמאחוריו ייתכן ונמצאות מחלקות רבות. הוא מגדיר ממשק ברמה גבוהה יותר במטרה לאפשר גישה נוחה לפונקציונליות המערכת.

ParkingLotFacade m = parkVehicle(Vehicle) ParkingTicket m unparkVehicle(Vehicle, PaymentTicket) void pay(ParkingTicket, PaymentMethod) PaymentTicket ParkingLot m = parkVehicle(Vehicle) ParkingTicket unparkVehicle(Vehicle, PaymentTicket) CarOwner pay(ParkingTicket, PaymentMethod) PaymentTicket parkVehicle(ParkingLotFacade) ParkingTicket generateParkingTicket(Vehicle, ParkingSpot) ParkingTicket handleFullParkingLotException(ParkingLotIsFullException) void generatePaymentTicket(ParkingTicket, double) PaymentTicket setLevelSelectionPolicy(LevelAssignmentPolicy) void pay(ParkingLotFacade, PaymentMethod) PaymentTicket setPaymentPolicy(PaymentCalculation) m unparkVehicle(ParkingLotFacade) void ParkingLevel getLevel(int) addLevel(ParkingLevel) void m = removeLevel(ParkingLevel) void

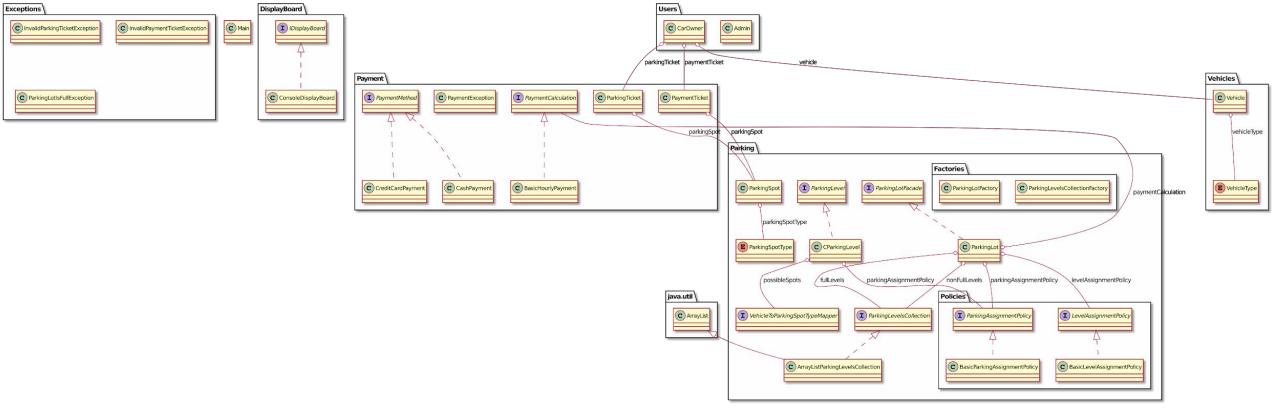
Facade Pattern

● בParkingLot ישנן מתודות כגון הוספת/מחיקת קומה/חניות שנרצה שרק האדמין יוכל לגשת אליהן.

כדי להימנע ממסירת כל הParkingLot לCarOwner ניצור שיועבר לFacade במקום כל הParkingLot.



PARKING LOT OOP's Class Diagram



PlantUML diagram generated by Sketchlit (https://bitbucket.org/pmesmeur/sketch.it)
For more information about this tool, please contact philippe.mesmeur@gmail.com