

הנדון: My Design Pattern

מחלקה: הנדסת תוכנה

פרטי המגישים:

* ספיר גילני 322358284
* עידן נוישול 207813635
* רועי דור 207813379

קורס תיכון מונחה עצמים 10119

My Design Pattern

# תוכן עניינים

[1. תוכן עניינים 2](#_Toc164789710)

[2. טבלת Design Pattern 3](#_Toc164789711)

[3. סיבות השימוש בDesign Pattern 5](#_Toc164789712)

[3.1 Factory 5](#_Toc164789716)

[3.2 Adapter 5](#_Toc164789717)

[3.3 Facade 5](#_Toc164789718)

[3.4 Command 5](#_Toc164789719)

[3.5 Observer 5](#_Toc164789720)

[3.6 Memento 5](#_Toc164789721)

[3.7 Singleton 5](#_Toc164789722)

[4. הערות 6](#_Toc164789723)

# טבלת Design Pattern

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מספר סידורי | תבנית | מחלקות | קבצים | סעיפים |
| 1 | Factory | ProductFactory | ProductFactory.java | 2.1, 2.2, 2.3 |
| יצירת מחלקה האחראית על יצירת אובייקטים המקיימים פולימורפיזם למוצר. | | |
| ShippingFactory | ShippingFactory.java | 3, 3.1 |
| יצירת מחלקה האחראית על יצירת אובייקטים המקיימים פולימורפיזם לחברת משלוחים. | | |
| InvoiceAdapterFactory | InvoiceAdapterFactory.java | 2.2, 2.3 |
| יצירת מחלקה האחראית על יצירת ממשקים המתממשקים לסוגי קבלות שונות. | | |
| 2 | Adapter | InvoiceAdapterFactory | InvoiceAdapterFactory.java | 2 |
| AccountantInvoice | AccountantInvoice.java | 2.2.2 |
| AccountantInvoiceAdapter | AccountantInvoiceAdapter.java | 2.2.2 |
| יצירת מעטפת לקבלה מסוג רואה חשבון העוטפת את פונקציית הממשק - הצגת החשבונית. | | |
| CustomerInvoice | Customer Invoice.java | 2.2.1 |
| CustomerInvoiceAdapter | CustomerInvoiceAdapter.java | 2.2.1 |
| יצירת מעטפת לקבלה מסוג לקוח העוטפת את פונקציית הממשק - הצגת החשבונית. | | |
| 3 | Facade | SystemFacade | SystemFacade.java | 4.11 |
| יצירת מעטפת ונקודת גישה מרכזית המפעילה את כלל הפעולות האפשריות במערכת. | | |
| 4 | Command | ICommand | ICommand.java | 3 |
| ממשק זה בעל שתי פעולות:   1. execute 2. עדכון מוצר | | |
| IShippingReceiver | IShippingReciever.java | 3 |
| ממשק המאפשר קבלת מידע אודות המשלוח | | |
| FedExExpressCommand | FedExExpressCommand.java | 3.1.1 |
| מחלקה המתממשקת לממשק הICommand ומפעילה את חישוב דמי המשלוח עבור חברת FedEx וסוג Express. | | |
| FedExStandardCommand | FedExStandardCommand.java | 3.1.2 |
| מחלקה המתממשקת לממשק הICommand ומפעילה את חישוב דמי המשלוח עבור חברת FedEx וסוג Standard. | | |
| DHLExpressCommand | DHLExpressCommand.java | 3.1.1 |
| מחלקה המתממשקת לממשק הICommand ומפעילה את חישוב דמי המשלוח עבור חברת DHL וסוג Express. | | |
| DHLStandardCommand | DHLStandardCommand.java | 3.1.2 |
| מחלקה המתממשקת לממשק הICommand ומפעילה את חישוב דמי המשלוח עבור חברת DHL וסוג Standard. | | |
| ShippingInvoker | ShippingInvoker.java | 3 |
| מחלקת השליטה שאחראית על ביצוע חישוב דמי משלוח של כל חברות המשלוח. | | |
| IUndoCommand | IUndoCommand.java | 4.6 |
| ממשק המאפשר פעולת undo | | |
| OrderUpdateCommand | OrderUpdateCommand.java | 4.6 |
| מחלקה השומרת את נתוני ההזמנה הקודמת ומוסיפה הזמנה למוצר. | | |
| OrderController | OrderController.java | 4.6 |
| מחלקת בקרה המחזיקה במחסנית של ממשקי undo להזמנות. | | |
| 5 | Observer | IObserver | IObserver.java | 3 |
| ממשק המאפשר את פעולת הupdate. | | |
| ObserverManagement | ObserverManagement.java | 3 |
| מחלקת בקרה לניהול והוספה של "מאזינים" המאפשרת שליחת הודעה לכלל ה"מאזינים" (חברות המשלוח). | | |
| ShippingCompany | ShippingCompany.java | 3 |
| מחלקה אבסטרקטית המתממשקת לממשק IObserver והמחלקות היורשות ממנה ממשות את הפעולה לשליחת הודעה אוטומטית. | | |
| 6 | Memento | Product | Product.java | 4.10 |
| ProductMemento | Product.java | 4.10 |
| יצירת גיבוי למוצר. | | |
| OrderController | OrderController.java | 4.10 |
| OrderControllerMemento | OrderController.java | 4.10 |
| יצירת גיבוי למחסנית ההזמנות. | | |
| Memento | SystemFacade.java | 4.10 |
| יצירת גיבוי למערכת. | | |
| 7 | Singleton | System Facade | SystemFacade.java | 4.11 |
| יצירת נקודת גישה יחידה וגלובלית למערכת שלא ניתן ליצור מופעים נוספים. | | |

# סיבות השימוש בDesign Pattern



## Factory

השימוש בFactory מאפשר לנו ליצור את האובייקט הפולימורפי ללא שימוש בcasting ויתרה מכך לוגיקת יצירת האובייקט מתבצעת במקום יחיד במערכת.

## Adapter

שימוש בAdapter מהווה כמעטפת לפעולה הרצויה, דבר זה אינו פוגע במערכת במידה ובפיתוח עתידי, ישתנה שם המחלקה/הפעולה.

## Facade

השימוש בFacade מרכז את כל הלוגיקה והפעולות של המערכת במקום אחד, דבר זה מאפשר לתוכנית הראשית, להשתמש בפעולות הקשורות למערכת.

## Command

אנו השתמשנו בCommand לפתירת שתי בעיות מרכזיות:

### Shipping

כחלק מדרישות הפרויקט, נדרשנו לתמוך ביצירת פעולה המחשבת עלות דמי משלוח, עם זאת לכל חברה יש סוג משלוח שונה המחשב באופן שונה את עלות המשלוח.   
לתמיכה בפעולה מסוג זה החלטנו כי האופן היעיל ביותר יהיה שימוש בCommand. לשם כך, יצרנו מחלקה המהווה "כמחלקת שליטה" המפעילה את כל חברות המשלוח השונות לחישוב משלוח מהסוג הנתון ומחזירה את החברת משלוח המציעה את העלות הזולה ביותר.

### Undo

כחלק מדרישת הפרויקט, נדרשנו לתמוך בפעולת undo.   
לתמיכה בפעולה מסוג זה החלטנו כי השימוש בCommand יאפשר את התמיכה הנכונה ביותר לנדרש ואף להרחבות עתידיות. לצורך המימוש נעזרנו במבנה נתונים מסוג מחסנית המחזיקה אובייקטים המממשים את הממשק IUndoCommand.

## Observer

לפי סעיף 3, התבקשנו לאפשר הודעה אוטומטית של האתר של החנות לכל חברות המשלוח השונות. בכדי לממש את הפעולה באופן היעיל ביותר, השתמשנו בObserver. במערכת ישנה מחלקת ניהול של ה"מאזינים", מחלקה זו מחזיקה הודעה וגם מבנה נתונים מסוג HashSet לשמירת ה"מאזינים". המחלקה יכולה להוסיף ולהוריד "מאזין" ובעלת הפעולה myNotify() אשר מבצעת את שליחת ההודעה האוטומטית.

## Memento

לפי סעיף 4.10, התבקשנו לאפשר גיבוי ושחזור למערכת. למימוש פעולות אלו, השתמשנו בMemento אשר מומש כמחלקה פנימית של המערכת. כמו כן, לכל תכונה רלוונטית מימשנו Memento עבורו.

## Singleton

לפי סעיף 4.11, התבקשנו שנקודת הגישה למערכת תהא יחידה ושלא יהיה ניתן ליצור מופעים נוספים ממנה. לצורך מימוש הפעולה, קבענו כי Facade המערכת יממש Singleton ובכך לא יהיה ניתן ליצור מופעים נוספים.

# הערות

1. לפי סעיף 2, צויין כי המערכת נדרשת לתמוך בסוגי משלוחים עתידיים. על פי הטבלה הנ"ל, את ניהול המשלוחים ביצענו בעזרת design pattern מסוג command. דבר זה מאפשר הרחבה בקלות ליצירת סוגי משלוחים חדשים (למשל משלוח מסוג אונייה).
2. לפי סעיף 3.1.1 צויין כי חברת DHL בעלת מס ייבוא שונה לכל מדינה. בפרויקט זה, התבקשנו לתת מס ייבוא אחיד לכולם של 20$. בכדי לתמוך במחירים שונים לכל מדינה, יש ליצור אוסף מסוג HashMap אשר המפתחות הינם מדינות והערכים הינם מס ייבוא של כל מדינה. ובעזרת המשתנה destCountry אשר משוייך לכל הזמנה החברות ייחשבו את דמי המשלוח לפי מס הייבוא המתאים.