

Design Patterns

מס'	שם התבנית	הסעיפים הרלוונטיים	מחלקות מממשות	סיבת השימוש
1	Singleton	4.11	מחלקה Store	לאורך התוכנית אנו נדרשים לגשת לשדות שונים שהישות של החנות מחזיקה, ללא דרך להגיע אליהם מבלי להעמיס על הפונקציות והמחלקות השונות. השימוש בתבנית זו מאפשר גישה לשדות החנות באופן פשוט. בנוסף לכך, אנחנו רוצים למנוע מצב שבו יהיה שיכפול של החנות, כדי שלא יהיה אי-סדר באוסף המוצרים וההזמנות במערכת.
2	Facade	4, 3, 2	מחלקה Store, מחלקה StorageManager, מחלקה ShippingManager, מחלקה OrderManager, כל מחלקות ה-Command השונות.	תבנית זו מאפשרת יצירת סדר על ידי איגוד תחומי אחריות במחלקה אחת. בעזרת השימוש בתבנית זו, יצרנו מערכת בה יש צורה ברורה לצורך גישה למחלקות השונות ולמתודות שלהן.
3	Iterator	4, 3, 2, 1	מחלקה Product, מחלקה Order, מחלקה StorageManager, מחלקה ShippingManager, מחלקה OrderManager.	תבנית זו מובנית בכל אוסף בו השתמשנו, ומאפשרת לעבור על איברים האוסף.
4	Memento	4.10	מחלקה BackupCommand, מחלקה RestoreCommand.	התבנית מאפשרת גיבוי של נתונים לפי בחירתנו בכל נקודת זמן. בהתאם לדרישה בסעיף המצוין, בחרנו להשתמש בתבנית זו כדי לגבות את אוסף המוצרים וההזמנות הקיימים בחנות, נכון לנקודה בה המשתמש בוחר לגבות. לאחר יצירת גיבוי, המשתמש יכול לבחור לשחזר את המערכת על סמך הגיבוי שיצר.
5	Observer עם שינויים	4.5	מחלקה CreateOrderCommand, מחלקה ShippingManager.	באופן הרגיל, התבנית מאפשרת למחלקות המממשות מאזינים לקבל עדכון כאשר אירוע כלשהו מתרחש במחלקה אחרת המממשת מודיע. במערכת שלנו המחלקה ShippingManager מקבלת התראה כאשר נוצרת הזמנה חדשה למוצר שנמכר דרך האתר על ידי המחלקה CreateOrderCommand. השוני אצלנו מהצורה הרגילה בה מממשים את התבנית, היא שהמאזין אצלנו מחזיר אובייקט ומודיע. במקרה הזה מדובר במחלקה המחזירה למודיע מידע על חברת המשלוחים המציעה את השירות הזול ביותר, ואת המחיר של המשלוח שלהם.
6	Command	4	בכל מחלקות ה-Command.	התבנית מאפשרת למחלקת הפקודה לאגד בתוכה את כל השדות והפונקציות הנדרשים לביצוע הפעולה לה היא מיועדת. בתוכנית שלנו בחרנו לממש את התבנית בתפריט הראשי, כך שכל פעולה שהמשתמש יכול לבצע, מתבצעת על ידי פקודה אחרת.