

## Classes

מס'	מחלקה וקיום עקרונות ה-SOLID
1	<p><u>CheapestShippingService</u> :</p> <p>S – אחריות המחלקה לאגד מידע לגבי חברת משלוחים ומחיר המשלוח שהם מציעים.</p> <p>O – שדות המחלקה נגישים לקריאה לכל מחלקה חיצונית (בפרט למחלקות יורשות). כל מחלקה יורשת יכולה בקלות להרחיב את הפונקציונליות של המתודות.</p> <p>L – המחלקה לא יורשת מאף מחלקה אחרת, ואף מחלקה לא יורשת ממנה, לכן היא מקיימת את העקרון במובן הריק.</p> <p>I – מקיימת את העקרון במובן הריק, שכן היא לא מממשת שום ממשק.</p> <p>D – המחלקה לא מחזיקה טיפוס ספציפי של חברת משלוחים, אלא את הטיפוס האבסטרקטי שמייצג את כל חברות המשלוחים.</p>
2	<p><u>Customer</u> :</p> <p>S – אחריות המחלקה להחזיק מידע על לקוח, ולאפשר לגשת למידע הזה.</p> <p>O – המחלקה רק מנגישה מידע, לכן זה יהיה פשוט להרחיב את פונקציונליות המחלקה.</p> <p>L – המחלקה לא יורשת מאף מחלקה אחרת, לכן היא מקיימת את העקרון במובן הריק.</p> <p>I – המחלקה מממשת את הממשק Cloneable המאפשר שימוש בפונקציה clone. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>D – המחלקה מחזיקה רק טיפוסים פרימיטיביים ולא תלויה במחלקות קונקרטיות העלולות להשתנות בעתיד.</p>
3	<p><u>DHL</u> :</p> <p>S – אחריות המחלקה לייצג חברת משלוחים ולהנגיש מידע על תעריפי המשלוחים שלה.</p> <p>O – חברה הרוצה להרחיב את המחלקה הזו וגם לחשב את תעריפי המשלוח באופן שונה יכולה לדרוס את המתודות של המחלקה ולממש אותן איך שהיא צריכה.</p> <p>L – המחלקה שומרת על הלוגיקה שמכתיבה לה מחלקת האב שלה, ShippingService.</p> <p>I – המחלקה מממשת את הממשק Cloneable המאפשר שימוש בפונקציה clone. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>D – המחלקה לא תלויה במימוש המתודות של ProductWebsite לכן כל מחלקה שתירש ממנה, תיתמך על ידי DHL.</p>
4	<p><u>FedEx</u> :</p> <p>S – אחריות המחלקה לייצג חברת משלוחים ולהנגיש מידע על תעריפי המשלוחים שלה.</p> <p>O – חברה הרוצה להרחיב את המחלקה הזו וגם לחשב את תעריפי המשלוח באופן שונה יכולה לדרוס את המתודות של המחלקה ולממש אותן איך שהיא צריכה.</p> <p>L – המחלקה שומרת על הלוגיקה שמכתיבה לה מחלקת האב שלה, ShippingService.</p> <p>I – המחלקה מממשת את הממשק Cloneable המאפשר שימוש בפונקציה clone. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>D – המחלקה לא תלויה במימוש המתודות של ProductWebsite לכן כל מחלקה שתירש ממנה, תיתמך על ידי FedEx.</p>
5	<p><u>Informer</u> : המחלקה מהווה חלק ממימוש של תבנית Observer עם מספר שינויים בכדי להתאים לפרויקט.</p> <p>S – אחריות המחלקה להיות תבנית למחלקה שצריכה להודיע למחלקות אחרות על פעולות שהיא מבצעת.</p> <p>O – כל מחלקה שצריכה לממש תבנית זו יכולה להרחיב על הלוגיקה שהתבנית מאפשרת ללא צורך לשנות את התבנית עצמה.</p> <p>L – המחלקה לא יורשת מאף מחלקה אחרת, ולכן מקיימת את העקרון במובן הריק.</p> <p>I – המחלקה לא מממשת אף ממשק, ולכן מקיימת את העקרון במובן הריק.</p> <p>D – המחלקה תלויה בצורה בה הממשק Informed כתוב, אך מדובר בגורם אבסטרקטי ולא מחלקה קונקרטית המממשת את הממשק.</p>
6	<p><u>Informed</u> : הממשק מהווה חלק ממימוש של תבנית Observer עם מספר שינויים בכדי להתאים לפרויקט.</p> <p>S – אחריות הממשק לקבל הודעה על פעולה כלשהי שמחלקה שיורשת מהמחלקה Informer ביצעה ולהניע פעולות משלו בתגובה.</p> <p>O – כל מחלקה המממשת את הממשק יכולה לקבוע את הפונקציונליות של המתודה שהממשק מספק.</p> <p>L – מדובר בממשק, לכן הוא מקיים את העקרון במובן הריק.</p> <p>I – הממשק מגדיר מתודה אחת בלבד, לכן כל מחלקה שתממש אותו ככל הנראה תצטרך את המתודה הזו.</p> <p>D – הממשק לא תלוי במימוש של אף מחלקה אחרת.</p>
7	<p><u>Invoice</u> :</p> <p>S – אחריות המחלקה לאגד מידע הנדרש להצגת חשבונית ללקוח.</p> <p>O – המחלקה מאפשרת למחלקות היורשות ממנה לקבוע את אופן המימוש של המתודות שלה.</p> <p>L – המחלקה לא יורשת מאף מחלקה אחרת, ולכן מקיימת את העקרון במובן הריק.</p> <p>I – המחלקה מממשת את הממשק Cloneable המאפשר שימוש בפונקציה clone. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p>

	<p>D – המחלקה לא תלויה במימוש של אף מחלקה אחרת. כל השדות שלה הם משתנים מטיפוסים פרימיטיביים או טיפוסים מוגדרים היטב.</p>
8	<p><u>InvoiceAccountant</u>:</p> <p>S – אחריות המחלקה לאגד ולהנגיש מידע רלוונטי לקבלה על הזמנה שמיועדת לרואה חשבון.</p> <p>O – המחלקה מגדירה את מימוש המתודות שלה, אך במידה ומחלקה תרצה להוסיף על המידע שהיא מייצגת, באפשרותה לרשת ממנה ולהרחיב את הפונקציונליות שלה.</p> <p>L – המחלקה שומרת על הלוגיקה שמחלקת האב שלה קובעת, ולא משנה או מוסיפה למתודה toString שלה פונקציונליות שאין או שהולכת נגד מחלקת האב שלה.</p> <p>I – המחלקה מממשת את הממשק Cloneable המאפשר שימוש בפונקציה clone. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>D – המחלקה תלויה רק במחלקת האב שלה ממנה היא יורשת.</p>
9	<p><u>InvoiceCustomer</u>:</p> <p>S – אחריות המחלקה לאגד ולהנגיש מידע רלוונטי לקבלה על הזמנה שמיועדת לרואה חשבון.</p> <p>O – המחלקה מגדירה את מימוש המתודות שלה, אך במידה ומחלקה תרצה להוסיף על המידע שהיא מייצגת, באפשרותה לרשת ממנה ולהרחיב את הפונקציונליות שלה.</p> <p>L – המחלקה שומרת על הלוגיקה שמחלקת האב שלה קובעת, ולא משנה או מוסיפה למתודה toString שלה פונקציונליות שאין או שהולכת נגד מחלקת האב שלה.</p> <p>I – המחלקה מממשת את הממשק Cloneable המאפשר שימוש בפונקציה clone. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>D – המחלקה תלויה רק במחלקת האב שלה ממנה היא יורשת.</p>
10	<p><u>Order</u>:</p> <p>S – אחריות המחלקה לאגד מידע על הזמנה שנוצרה עבור לקוח. המידע כולל מזהה של מוצר, הכמות שלו, מזהה של ההזמנה, ואת פרטי הלקוח המזמין.</p> <p>O – המחלקה מאפשרת ירושה לצורך הרחבת או מיקוד הפונקציונליות שלה. המחלקה עצמה מספקת גישה לשדות שלה, ומחלקה יורשת יכולה להוסיף פונקציונליות לפי צורכיה.</p> <p>L – המחלקה לא יורשת מאף מחלקה אחרת, ולכן מקיימת את העקרון במובן הריק.</p> <p>I – המחלקה מממשת את הממשק Cloneable המאפשר שימוש בפונקציה clone. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>D – המחלקה לא תלויה בשום מימוש ספציפי של המחלקות עליהן היא מתבססת, כגון Customer.</p>
11	<p><u>OrderManager</u>:</p> <p>S – אחריות המחלקה ליצור ולתחזק את אוסף ההזמנות של החנות.</p> <p>O – המחלקה מאפשרת ירושה ללא צורך בשינוי הקוד שלה. מתודות תחזוקת אוסף ההזמנות לא מוגבלות לסוגים ספציפיים של הזמנה.</p> <p>L – המחלקה לא יורשת מאף מחלקה אחרת, ולכן מקיימת את העקרון במובן הריק.</p> <p>I – המחלקה מממשת את הממשק Cloneable המאפשר שימוש בפונקציה clone. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>D – המחלקה לא תלויה בטיפוס קונקרטי של הזמנה ועובדת עם המחלקה האבסטרקטית.</p>
12	<p><u>OrderShipped</u>:</p> <p>S – אחריות המחלקה לאגד ולהנגיש מידע על הזמנה שאמורה להישלח ללקוח.</p> <p>O – המחלקה מאפשרת ירושה לצורך הרחבת הפונקציונליות שלה ללא צורך בשכתוב הקוד שלה.</p> <p>L – המחלקה לא משנה את מימוש המתודות שהיא יורשת ממחלקת האב שלה.</p> <p>I – המחלקה מממשת את הממשק Cloneable המאפשר שימוש בפונקציה clone. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>D – המחלקה לא תלויה בחברת משלוחים או בסוג מוצר ספציפיים. היא מתייחסת למחלקות האבסטרקטיות שלהם.</p>
13	<p><u>OrderStore</u>:</p> <p>S – אחריות המחלקה לאגד ולהנגיש מידע על הזמנה של מוצר שנמכר בחנות.</p> <p>O – המחלקה מאפשרת ירושה לצורך הרחבת הפונקציונליות שלה ללא צורך בשכתוב הקוד שלה.</p> <p>L – המחלקה לא משנה את מימוש המתודות שהיא יורשת ממחלקת האב שלה.</p> <p>I – המחלקה מממשת את הממשק Cloneable המאפשר שימוש בפונקציה clone. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>D – המחלקה לא משתמשת בלוגיקה שמממשות מחלקות הקבלה שהיא מחזיקה, לכן היא לא תלויה במימוש שלהן.</p>
14	<p><u>OrderWholeSaler</u>:</p> <p>S – אחריות המחלקה לאגד ולהנגיש מידע על הזמנה של מוצר שנמכר לסיטונאים.</p> <p>O – המחלקה מאפשרת ירושה לצורך הרחבת הפונקציונליות שלה ללא צורך בשכתוב הקוד שלה.</p> <p>L – המחלקה לא משנה את מימוש המתודות שהיא יורשת ממחלקת האב שלה.</p> <p>I – המחלקה מממשת את הממשק Cloneable המאפשר שימוש בפונקציה clone. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p>

	D – המחלקה לא משתמשת בלוגיקה שמממשת מחלקת הקבלה שהיא מחזיקה, לכן היא לא תלויה במימוש שלה.
15	<p><u>:Product</u></p> <p>S – אחריות המחלקה לאגד ולהנגיש מידע המייצג מוצר כללי שהחנות מוכרת.</p> <p>O – המחלקה מאפשרת ירושה לצורך הרחבה או מיקוד הפונקציונליות שלה ללא צורך בשכתוב הקוד שלה.</p> <p>L – המחלקה לא יורשת מאף מחלקה אחרת, ולכן מקיימת את העקרון במובן הריק.</p> <p>I – המחלקה מממשת את הממשק Cloneable המאפשר שימוש בפונקציה clone. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>D – המחלקה לא תלויה במימוש של מחלקות אחרות בהן היא משתמשת.</p>
16	<p><u>:ProductComparator</u></p> <p>S – אחריות המחלקה להשוות בין שני מוצרים על בסיס המק"ט שלהם.</p> <p>O – המחלקה לא תומכת בהרחבה שכן יש לה תפקיד ספציפי לבצע, וכל הרחבה שלה ככל הנראה יגרום לסטייה מימוש התפקיד הזה.</p> <p>L – המחלקה לא יורשת מאף מחלקה אחרת, ולכן מקיימת את העקרון במובן הריק.</p> <p>I – המחלקה מממשת את הממשק &lt;Product&gt;Comparator המאפשר שימוש בפונקציה compare. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>D – המחלקה לא תלויה במימוש ספציפי של Product שכן היא משתמשת במחלקה אבסטרקטית.</p>
17	<p><u>:ProductInStore</u></p> <p>S – אחריות המחלקה לייצג מוצר שנמכר פיזית בחנות ולהנגיש את המידע של המוצר למחלקות אחרות.</p> <p>O – המחלקה מאפשרת הוספת פונקציונליות ללא צורך בשכתוב הקוד שלה.</p> <p>L – המחלקה שומרת על הלוגיקה של מחלקת האב שלה ולא משנה או מוסיפה פונקציונליות למתודות של מחלקת האב שלה שהיא דורסת.</p> <p>I – המחלקה מממשת את הממשק Cloneable המאפשר שימוש בפונקציה clone. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>D – המחלקה לא תלויה במימוש של אף מחלקה מעבר למחלקת האב שלה.</p>
18	<p><u>:ProductWebsite</u></p> <p>S – אחריות המחלקה לייצג מוצר שנמכר דרך אתר החנות ולהנגיש את המידע של המוצר למחלקות אחרות.</p> <p>O – המחלקה מאפשרת הוספת פונקציונליות ללא צורך בשכתוב הקוד שלה.</p> <p>L – המחלקה שומרת על הלוגיקה של מחלקת האב שלה ולא משנה או מוסיפה פונקציונליות למתודות של מחלקת האב שלה שהיא דורסת.</p> <p>I – המחלקה מממשת את הממשק Cloneable המאפשר שימוש בפונקציה clone. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>D – המחלקה לא תלויה במימוש של אף מחלקה מעבר למחלקת האב שלה.</p>
19	<p><u>:ProductWholesalers</u></p> <p>S – אחריות המחלקה לייצג מוצר שנמכר לסיטונאים ולהנגיש את המידע של המוצר למחלקות אחרות.</p> <p>O – המחלקה מאפשרת הוספת פונקציונליות ללא צורך בשכתוב הקוד שלה.</p> <p>L – המחלקה שומרת על הלוגיקה של מחלקת האב שלה ולא משנה או מוסיפה פונקציונליות למתודות של מחלקת האב שלה שהיא דורסת.</p> <p>I – המחלקה מממשת את הממשק Cloneable המאפשר שימוש בפונקציה clone. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>D – המחלקה לא תלויה במימוש של אף מחלקה מעבר למחלקת האב שלה.</p>
20	<p><u>:Program</u></p> <p>S – אחריות המחלקה לנהל את אינטראקציית המערכת מול המשתמש, על ידי קליטת הקלט מהמשתמש ושליחתו לפקודות הרלוונטיות לביצוע הפעולות שהמשתמש בחר.</p> <p>O – המחלקה תומכת בהרחבה שכן הוספת פעולות נוספות שהמערכת עלולה לאפשר לא תדרוש שכתוב משמעותי של הקוד של המחלקה.</p> <p>L – המחלקה לא יורשת מאף מחלקה אחרת, ולכן מקיימת את העקרון במובן הריק.</p> <p>I – המחלקה לא מממשת אף ממשק, ולכן מקיימת את העקרון במובן הריק.</p> <p>D – המחלקה לא תלויה במימוש המחלקות האחרות, שכן היא אחראית ליזום תהליכים שהמערכת מאפשרת.</p>
21	<p><u>:ShippingManager</u></p> <p>S – אחריות המחלקה ליצור ולתחזק את אוסף חברות המשלוחים במערכת.</p> <p>O – ניתן להוסיף למחלקה פונקציונליות ללא צורך לשכתב את הקוד הקיים שלה.</p> <p>L – המחלקה לא יורשת מאף מחלקה אחרת, ולכן מקיימת את העקרון במובן הריק.</p> <p>I – המחלקה מממשת שני ממשקים, כאשר כל אחד מהם אחראי לפעולה נפרדת. למחלקה אין מתודות שהיא מממשת ללא צורך.</p> <p>D – המחלקה לא תלויה במימוש של ShippingService, שכן היא מחזיקה אוסף שמבחינתה מכיל מחלקה אבסטרקטית. היא גם לא תלויה במימוש של ProductWebsite ויכולה לעבוד עם מחלקות שעלולות לרשת ממנה.</p>
22	<p><u>:ShippingService</u></p> <p>S – אחריות המחלקה להוות תבנית שחברת משלוחים תצטרך לעמוד בה.</p>

	<p>O – כל חברת משלוחים היורשת ממחלקה זו מממשת באופן המתאים לה את המתודות לחישוב תעריפי המשלוח ללא צורך לשנות את הקוד של המחלקה.</p> <p>L – המחלקה לא יורשת מאף מחלקה אחרת, ולכן מקיימת את העקרון במובן הריק.</p> <p>I – המחלקה מממשת את הממשק Cloneable המאפשר שימוש בפונקציה clone. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>D – המחלקה לא תלויה במימוש של ProductWebsite, אלא רק בשדות שהיא, וכל המחלקות שעוללות לרשת ממנה, מחזיקה.</p>
23	<p><u>StorageManager</u> :</p> <p>S – אחריות המחלקה ליצור ולתחזק את אוסף המוצרים בחנות.</p> <p>O – המחלקה מאפשרת הוספת פונקציונליות ללא פגיעה במתודות הקיימות שלה, וללא צורך לשכתב אותן.</p> <p>L – המחלקה לא יורשת מאף מחלקה אחרת, ולכן מקיימת את העקרון במובן הריק.</p> <p>I – המחלקה מממשת את הממשק Cloneable המאפשר שימוש בפונקציה clone. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>- D</p>
24	<p><u>Store</u> : המחלקה מממשת את התבנית Singleton</p> <p>S – אחריות המחלקה לאגד בתוכה את המחלקות השונות המנהלות את המערכת ולהנגיש אותן למחלקות חיצוניות.</p> <p>O – המחלקה לא ניתנת להרחבה מלאה, שכן הבנאי שלה פרטי ולכן מחלקות יורשות לא יוכלו להסתמך עליה ליצירת instances.</p> <p>L – המחלקה לא יורשת מאף מחלקה אחרת, ולכן מקיימת את העקרון במובן הריק.</p> <p>I – המחלקה לא מממשת אף ממשק, ולכן מקיימת את העקרון במובן הריק.</p> <p>D – המחלקה לא תלויה במימוש של טיפוסים השדות שלה, שכן היא רק מחזיקה אותם ומנגישה אותם למחלקות האחרות.</p>
25	<p><u>ICommand</u> : המחלקה מהווה ממשק למימוש תבנית Command</p> <p>S – הממשק מגדיר פונקציה יחידה שנדרשת למימוש התבנית.</p> <p>O – הממשק לא ניתן להרחבה, אחרת הוא לא יעמוד בדרישות התבנית.</p> <p>L – הממשק לא יורש מאף מחלקה אחרת, ולכן מקיים את העקרון במובן הריק.</p> <p>I – הממשק לא מממש אף ממשק אחר, ולכן מקיים את העקרון במובן הריק.</p> <p>D – הממשק לא תלוי באף מחלקה אחרת.</p>
26	<p><u>BackupCommand</u> : המחלקה מממשת את התבנית Command</p> <p>S – המחלקה אחראית לבצע את פעולת הגיבוי למערכת.</p> <p>O – המחלקה מאפשרת הרחבת הפונקציונליות שלה, אם כי זה יבוא כנגד העקרון הקודם שכן כרגע היא מכילה בדיוק את כל הכלים שהיא צריכה לביצוע תחום האחריות שלה.</p> <p>L – המחלקה לא שומרת על העקרון משום שכל Command מבצע לוגיקה אחרת בקריאה למתודות execute שלו. בכדי לממש את התבנית יש לעבור על עקרון זה.</p> <p>I – המחלקה מממשת את הממשק ICommand המאפשר שימוש בפונקציה execute. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>D – המחלקה לא תלויה במימוש של אף מחלקה אחרת.</p>
27	<p><u>CreateOrderCommand</u> : המחלקה מממשת את התבנית Command</p> <p>S – המחלקה אחראית לבצע את הפעולה של יצירת הזמנה למוצר. למטרה זו היא נדרשת לקרוא לכל המחלקות הרלוונטיות לביצוע הפעולה ומאגדת בתוכה מספר פעולות עם לוגיקה שונה, לכן היא לא מקיימת את העקרון.</p> <p>המחלקה גם מהווה Facade שכן היא מאגדת תקשורת עם מספר מחלקות.</p> <p>O – המחלקה מאפשרת הרחבת הפונקציונליות שלה, אם כי זה יבוא כנגד העקרון הקודם שכן כרגע היא מכילה בדיוק את כל הכלים שהיא צריכה לביצוע תחום האחריות שלה.</p> <p>L – המחלקה לא שומרת על העקרון משום שכל Command מבצע לוגיקה אחרת בקריאה למתודות execute שלו. בכדי לממש את התבנית יש לעבור על עקרון זה.</p> <p>I – המחלקה מממשת את הממשק ICommand המאפשר שימוש בפונקציה execute. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>D – בכך שאנו בודקים את ערכי השדות שאנו קולטים מן המשתמש, אנו נמנעים מהתלות במחלקה Program לבדיקת תקינות הקלט.</p>
28	<p><u>CreateProductCommand</u> : המחלקה מממשת את התבנית Command</p> <p>S – המחלקה אחראית לבצע את הפעולה של הוספת מוצר לחנות. למטרה זו היא נדרשת לקרוא לכל המחלקות הרלוונטיות לביצוע הפעולה ומאגדת בתוכה מספר פעולות עם לוגיקה שונה, לכן היא לא מקיימת את העקרון.</p> <p>המחלקה גם מהווה Facade שכן היא מאגדת תקשורת עם מספר מחלקות.</p> <p>O – המחלקה מאפשרת הרחבת הפונקציונליות שלה, אם כי זה יבוא כנגד העקרון הקודם שכן כרגע היא מכילה בדיוק את כל הכלים שהיא צריכה לביצוע תחום האחריות שלה.</p> <p>L – המחלקה לא שומרת על העקרון משום שכל Command מבצע לוגיקה אחרת בקריאה למתודות execute שלו. בכדי לממש את התבנית יש לעבור על עקרון זה.</p>

	<p>I – המחלקה מממשת את הממשק ICommand המאפשר שימוש בפונקציה execute. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>D – בכך שאנו בודקים את ערכי השדות שאנו קולטים מן המשתמש, אנו נמנעים מהתלות במחלקה Program לבדיקת תקינות הקלט.</p>
29	<p><u>PrintAllProductsCommand : המחלקה מממשת את התבנית Command</u></p> <p>S – המחלקה אחראית לבצע את הפעולה של הדפסת הפרטים של כל המוצרים וכל ההזמנות שיש לחנות. למטרה זו היא נדרשת לקרוא לכל המחלקות הרלוונטיות לביצוע הפעולה ומאגדת בתוכה מספר פעולות עם לוגיקה שונה, לכן היא לא מקיימת את העקרון. המחלקה גם מהווה Facade שכן היא מאגדת תקשורת עם מספר מחלקות.</p> <p>O – המחלקה מאפשרת הרחבת הפונקציונליות שלה, אם כי זה יבוא כנגד העקרון הקודם שכן כרגע היא מכילה בדיוק את כל הכלים שהיא צריכה לביצוע תחום האחריות שלה.</p> <p>L – המחלקה לא שומרת על העקרון משום שכל Command מבצע לוגיקה אחרת בקריאה למתודות execute שלו. בכדי לממש את התבנית יש לעבור על עקרון זה.</p> <p>I – המחלקה מממשת את הממשק ICommand המאפשר שימוש בפונקציה execute. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>D – המחלקה לא תלויה בקלט מהמשתמש ולא במימוש של מחלקות אחרות.</p>
30	<p><u>PrintOrdersForProductCommand : המחלקה מממשת את התבנית Command</u></p> <p>S – המחלקה אחראית לבצע את הפעולה של הדפסת כל ההזמנות של מוצר מסוים מהחנות. למטרה זו היא נדרשת לקרוא לכל המחלקות הרלוונטיות לביצוע הפעולה ומאגדת בתוכה מספר פעולות עם לוגיקה שונה, לכן היא לא מקיימת את העקרון. המחלקה גם מהווה Facade שכן היא מאגדת תקשורת עם מספר מחלקות.</p> <p>O – המחלקה מאפשרת הרחבת הפונקציונליות שלה, אם כי זה יבוא כנגד העקרון הקודם שכן כרגע היא מכילה בדיוק את כל הכלים שהיא צריכה לביצוע תחום האחריות שלה.</p> <p>L – המחלקה לא שומרת על העקרון משום שכל Command מבצע לוגיקה אחרת בקריאה למתודות execute שלו. בכדי לממש את התבנית יש לעבור על עקרון זה.</p> <p>I – המחלקה מממשת את הממשק ICommand המאפשר שימוש בפונקציה execute. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>D – בכך שאנו בודקים את ערכי השדות שאנו קולטים מן המשתמש, אנו נמנעים מהתלות במחלקה Program לבדיקת תקינות הקלט.</p>
31	<p><u>PrintProductCommand : המחלקה מממשת את התבנית Command</u></p> <p>S – המחלקה אחראית לבצע את הפעולה של הדפסת הפרטים של מוצר מסוים מהחנות וכל ההזמנות שלו. למטרה זו היא נדרשת לקרוא לכל המחלקות הרלוונטיות לביצוע הפעולה ומאגדת בתוכה מספר פעולות עם לוגיקה שונה, לכן היא לא מקיימת את העקרון. המחלקה גם מהווה Facade שכן היא מאגדת תקשורת עם מספר מחלקות.</p> <p>O – המחלקה מאפשרת הרחבת הפונקציונליות שלה, אם כי זה יבוא כנגד העקרון הקודם שכן כרגע היא מכילה בדיוק את כל הכלים שהיא צריכה לביצוע תחום האחריות שלה.</p> <p>L – המחלקה לא שומרת על העקרון משום שכל Command מבצע לוגיקה אחרת בקריאה למתודות execute שלו. בכדי לממש את התבנית יש לעבור על עקרון זה.</p> <p>I – המחלקה מממשת את הממשק ICommand המאפשר שימוש בפונקציה execute. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>D – בכך שאנו בודקים את ערכי השדות שאנו קולטים מן המשתמש, אנו נמנעים מהתלות במחלקה Program לבדיקת תקינות הקלט.</p>
32	<p><u>RemoveProductCommand : המחלקה מממשת את התבנית Command</u></p> <p>S – המחלקה אחראית לבצע את הפעולה של הסרת מוצר מהחנות. למטרה זו היא נדרשת לקרוא למספר מחלקות לביצוע הפעולה ומאגדת בתוכה מספר פעולות עם לוגיקה שונה, לכן היא לא מקיימת את העקרון. המחלקה גם מהווה Facade שכן היא מאגדת תקשורת עם מספר מחלקות.</p> <p>O – המחלקה מאפשרת הרחבת הפונקציונליות שלה, אם כי זה יבוא כנגד העקרון הקודם שכן כרגע היא מכילה בדיוק את כל הכלים שהיא צריכה לביצוע תחום האחריות שלה.</p> <p>L – המחלקה לא שומרת על העקרון משום שכל Command מבצע לוגיקה אחרת בקריאה למתודות execute שלו. בכדי לממש את התבנית יש לעבור על עקרון זה.</p> <p>I – המחלקה מממשת את הממשק ICommand המאפשר שימוש בפונקציה execute. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>D – בכך שאנו בודקים את ערכי השדות שאנו קולטים מן המשתמש, אנו נמנעים מהתלות במחלקה Program לבדיקת תקינות הקלט.</p>
33	<p><u>RestoreCommand : המחלקה מממשת את התבנית Command</u></p> <p>S – המחלקה אחראית לבצע את הפעולה של הדפסת הפרטים של מוצר מסוים מהחנות וכל ההזמנות שלו. למטרה זו היא נדרשת לקרוא לכל המחלקות הרלוונטיות לביצוע הפעולה ומאגדת בתוכה מספר פעולות עם לוגיקה שונה, לכן היא לא מקיימת את העקרון. המחלקה גם מהווה Facade שכן היא מאגדת תקשורת עם מספר מחלקות.</p> <p>O – המחלקה מאפשרת הרחבת הפונקציונליות שלה, אם כי זה יבוא כנגד העקרון הקודם שכן כרגע היא מכילה בדיוק את כל הכלים שהיא צריכה לביצוע תחום האחריות שלה.</p>

	<p>L – המחלקה לא שומרת על העקרון משום שכל Command מבצע לוגיקה אחרת בקריאה למתודת execute שלו. בכדי לממש את התבנית יש לעבור על עקרון זה.</p> <p>I – המחלקה מממשת את הממשק ICommand המאפשר שימוש בפונקציה execute. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>D – בכך שאנו בודקים את ערכי השדות שאנו קולטים מן המשתמש, אנו נמנעים מהתלות במחלקה Program לבדיקת תקינות הקלט.</p>
34	<p><u>SystemTestCommand : המחלקה מממשת את התבנית Command</u></p> <p>S – המחלקה אחראית לבצע את פעולת שחזור המערכת מגיבוי שיצרנו.</p> <p>O – המחלקה מאפשרת הרחבת הפונקציונליות שלה, אם כי זה יבוא כנגד העקרון הקודם שכן כרגע היא מכילה בדיוק את כל הכלים שהיא צריכה לביצוע תחום האחריות שלה.</p> <p>L – המחלקה לא שומרת על העקרון משום שכל Command מבצע לוגיקה אחרת בקריאה למתודת execute שלו. בכדי לממש את התבנית יש לעבור על עקרון זה.</p> <p>I – המחלקה מממשת את הממשק ICommand המאפשר שימוש בפונקציה execute. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>D – המחלקה לא תלויה במימוש של אף מחלקה אחרת.</p>
35	<p><u>UndoOrderCommand : המחלקה מממשת את התבנית Command</u></p> <p>S – המחלקה אחראית לבצע את הפעולה של הסרת ההזמנה האחרונה שנוצרה. למטרה זו היא נדרשת לקרוא לכל המחלקות הרלוונטיות לביצוע הפעולה ומאגדת בתוכה מספר פעולות עם לוגיקה שונה, לכן היא לא מקיימת את העקרון. המחלקה גם מהווה Facade שכן היא מאגדת תקשורת עם מספר מחלקות.</p> <p>O – המחלקה מאפשרת הרחבת הפונקציונליות שלה, אם כי זה יבוא כנגד העקרון הקודם שכן כרגע היא מכילה בדיוק את כל הכלים שהיא צריכה לביצוע תחום האחריות שלה.</p> <p>L – המחלקה לא שומרת על העקרון משום שכל Command מבצע לוגיקה אחרת בקריאה למתודת execute שלו. בכדי לממש את התבנית יש לעבור על עקרון זה.</p> <p>I – המחלקה מממשת את הממשק ICommand המאפשר שימוש בפונקציה execute. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>D – בכך שאנו בודקים את ערכי השדות שאנו קולטים מן המשתמש, אנו נמנעים מהתלות במחלקה Program לבדיקת תקינות הקלט.</p>
36	<p><u>UpdateQuantityCommand : המחלקה מממשת את התבנית Command</u></p> <p>S – המחלקה אחראית לבצע את הפעולה של עדכון מלאי של מוצר. למטרה זו היא נדרשת לקרוא למספר מחלקות לביצוע הפעולה ומאגדת בתוכה מספר פעולות עם לוגיקה שונה, לכן היא לא מקיימת את העקרון.</p> <p>O – המחלקה מאפשרת הרחבת הפונקציונליות שלה, אם כי זה יבוא כנגד העקרון הקודם שכן כרגע היא מכילה בדיוק את כל הכלים שהיא צריכה לביצוע תחום האחריות שלה.</p> <p>L – המחלקה לא שומרת על העקרון משום שכל Command מבצע לוגיקה אחרת בקריאה למתודת execute שלו. בכדי לממש את התבנית יש לעבור על עקרון זה.</p> <p>I – המחלקה מממשת את הממשק ICommand המאפשר שימוש בפונקציה execute. מעבר לכך, המחלקה לא מחזיקה מתודות שאין לה שימוש בהן.</p> <p>D – בכך שאנו בודקים את ערכי השדות שאנו קולטים מן המשתמש, אנו נמנעים מהתלות במחלקה Program לבדיקת תקינות הקלט.</p>