

המחלקה	Single Responsibility Principle	Open/Closed Principle	Liskov Substitution Principle	Interface Segregation Principle
Store	למחלקה יש אחריות אחת וזה לנהל את הנתונים שיש לחנות	המחלקה פתוחה להרחבות. עבור שינויים עתידיים, וסגורה לשינויים על ידי פונקציות המאפשרות שינוי מידע ללא צורך עריכה או שינוי שלהם.	מאפשרת הורשה במידת הצורך, נכון ליצירת הפרויקט לא היה בזה שימוש.	זוהי מחלקה המנהל את כל נתוני החנות, לכן לא מומש כאן עקרון ההפרדה.
Product	אחריות לייצוג של מוצר וניהול המאפיינים שלו, כמו שם, מחיר, מלאי, משקל והזמנות. היא מספקת שיטות לגישה ושינוי של מאפיינים אלה, כמו גם הוספה והסרה של הזמנות המשויכות למוצר.	המחלקה פתוחה להרחבה. ניתן להוסיף לה שיטות או התנהגויות חדשות כדי לשפר את הפונקציונליות שלה מבלי לשנות את הקוד הקיים. לדוגמה, ניתן להוסיף שיטות לחישוב הכנסות, בדיקת זמינות או עדכון מחירים מבלי לשנות את הליבה של המחלקה.	נראה ש-Product עומדת בעקרון LSP על ידי מתן ממשק עקבי לעבודה עם מוצרים. ניתן להחליף מופעים של Product במופעים של תת-מחלקות שלה מבלי להשפיע על אופן השימוש שלהן על ידי מחלקות אחרות.	המחלקה לא מממשת אף ממשק, אבל היא מספקת קבוצת שיטות קוהרנטיות שקשורות לניהול נתוני מוצר והזמנות. לקוחות יכולים ליצור אינטראקציה עם שיטות אלה בהתאם לצרכים שלהם מבלי להיות תלויים בפונקציונליות מיותרת.
Person	למחלקת Person יש אחריות יחידה - לייצג אדם עם שם ומספר נייד. היא מספקת שיטות לקבלת והגדרת השם והמספר הנייד, וכן שיטה לאימות מספר הנייד. כל השיטות הללו קשורות לניהול הנתונים המשויכים לאדם.	המחלקה פתוחה להרחבה. ניתן ליצור תת-מחלקות שלה כדי להוסיף פונקציונליות חדשה הקשורה לאנשים. לדוגמה, ניתן ליצור תת-מחלקות כדי לייצג סוגים שונים של אנשים, כגון לקוחות, עובדים או ספקים. עם זאת, המחלקה עצמה סגורה לשינוי, מכיוון שלא צריך לשנות את הפונקציונליות הקיימת שלה כדי להתאים לסוגים חדשים של אנשים.	מאפשרת ממשק עקבי לניהול של אנשים, כאשר כל מחלקה יורשת יכולה להוסיף מימוש מבלי להשפיע על מחלקה זו.	ללקוחות יש אפשרות ליצור אינטראקציה עם שיטות ספציפיות הרלוונטיות לצרכים שלהם מבלי להיות תלויים בפונקציונליות מיותרת. ובכך בעקיפין מממשת את עקרון זה.
Order	מחלקת ההזמנה (Order) אחראית לייצוג של הזמנה וניהול המאפיינים שלה, כגון מוצר, לקוח, כמות וסוג משלוח. היא גם מחשבת את הסכום הכולל של ההזמנה ומספקת שיטה ליצירת חשבונית. נראה שלמחלקה יש אחריות יחידה הקשורה לניהול הזמנות.	המחלקה פתוחה להרחבה. ניתן להוסיף לה שיטות או התנהגויות חדשות כדי לשפר את הפונקציונליות שלה מבלי לשנות את הקוד הקיים. לדוגמה, ניתן להוסיף שיטות לטיפול בהנחות מיוחדות, מעקב אחר סטטוס ההזמנה או אימות פרטי ההזמנה מבלי לשנות את הליבה של המחלקה.	המחלקה מספקת ממשק עקבי לעבודה עם הזמנות, ללא קשר לסוג הספציפי של ההזמנה.	המחלקה לא מממשת אף ממשק
SoldInStore	למחלקת SoldInStore יש אחריות יחידה - לייצג מוצרים שנמכרים בחנות ולספק מידע לגבי התמחור שלהם.	המחלקה פתוחה להרחבה אך סגורה לשינוי. ניתן להרחיב את הפונקציונליות שלה על ידי הוספת שיטות או מאפיינים חדשים, אך ההתנהגות הקיימת שלה נשארת ללא שינוי.	ניתן להשתמש במופעים של SoldInStore באופן זהה למופעים של מחלקת Product, מכיוון ש-SoldInStore יורשת מ-Product ומורחבת את הפונקציונליות שלה מבלי לשנות את אופן הפעולה שלה.	המחלקה לא מממשת אף ממשק
SoldToWholesalers	מייצגת מוצרים שנמכרים לסיטונאים	המחלקה פתוחה להרחבה. ניתן להרחיב אותה כדי להוסיף עוד פונקציונליות או התנהגות הקשורות למוצרים שנמכרים לסיטונאים. עם זאת, היא גם סגורה לשינוי, מכיוון שההתנהגות הקיימת שלה יכולה להיות מנוצלת חזרה מבלי לשנות את המחלקה עצמה.	ניתן להשתמש במופעים של SoldToWholesalers באופן זהה למופעים של מחלקת Product, מכיוון ש-SoldToWholesalers יורשת מ-Product ומורחבת את הפונקציונליות שלה מבלי לשנות את אופן הפעולה שלה.	המחלקה מספקת ממשק קוהרנטי על ידי התמקדות בייצוג מוצרים שנמכרים לסיטונאים.
SoldTroughWebsite	מייצגת מוצרים שנמכרים דרך אתר אינטרנט. היא מטפלת בפונקציונליות הקשורה להזמנות שבוצעו דרך האתר ומספקת ייצוג מחרוזת של מוצרים אלה, כולל פרטים כמו ארץ יעד וסה"כ מחיר מכירה.	פתוחה להרחבה. ניתן SoldTroughWebsite להרחיב אותה כדי להוסיף עוד פונקציונליות או התנהגות הקשורות למוצרים שנמכרים דרך האתר. היא גם סגורה לשינוי, מכיוון שההתנהגות הקיימת שלה יכולה להיות מנוצלת חזרה מבלי לשנות את המחלקה עצמה.	ניתן להשתמש במופעים של SoldTroughWebsite באופן זהה למופעים של מחלקת Product, מכיוון ש-SoldTroughWebsite יורשת מ-Product ומורחבת את הפונקציונליות שלה מבלי לשנות את אופן הפעולה שלה.	המחלקה מספקת ממשק קוהרנטי על ידי התמקדות בייצוג מוצרים שנמכרים דרך אתר אינטרנט. לקוחות יכולים ליצור אינטראקציה עם השיטות שלה כדי לאחזר מידע לגבי הזמנות שבוצעו דרך האתר, מבלי להיות תלויים בפונקציונליות מיותרת.

Customer	למחלקת Customer יש אחריות יחידה - לייצג לקוח. היא מורישה מהמחלקה Person כדי לרשת מאפיינים בסיסיים כמו שם ומספר נייד ומוסיפה התנהגות ספציפית הקשורה ללקוחות, כגון יצירת מחרוזת המייצגת לקוח.	המחלקה פתוחה להרחבה. ניתן להרחיב אותה כדי לכלול פונקציונליות נוספת הקשורה ללקוחות מבלי לשנות את ההתנהגות הקיימת שלה. היא גם סגורה לשינוי, מכיוון שההתנהגות הנוכחית שלה יכולה להיות מנוצלת חזרה על ידי משתמשים.	ניתן להשתמש במופעים של Customer באופן זהה למופעים של Person, מכיוון ש-Customer יורשת מ-Person ומורחבת את הפונקציונליות שלה מבלי לשנות את אופן הפעולה שלה. משמעות הדבר היא שבכל מקום שמצפים ל-Person, ניתן להשתמש ב-Customer ללא בעיות.	המחלקה מספקת ממשק קוהרנטי על ידי התמקדות בייצוג סיטונאים. היא לא חושפת שיטות או התנהגויות מיותרות, ושומרת על הממשק שלה נקי ותמציתי.
Wholesaler	למחלקת Wholesaler יש אחריות יחידה - לייצג סיטונאי. היא יורשת מהמחלקה Person כדי לרשת מאפיינים בסיסיים כמו שם ומספר נייד ומוסיפה התנהגות ספציפית הקשורה לסיטונאים, כגון יצירת מחרוזת המייצגת סיטונאי.	המחלקה פתוחה להרחבה. ניתן להרחיב אותה כדי לכלול פונקציונליות נוספת הקשורה לסיטונאים מבלי לשנות את ההתנהגות הקיימת שלה. היא גם סגורה לשינוי, מכיוון שההתנהגות הנוכחית שלה יכולה להיות מנוצלת חזרה על ידי המשתמש מבלי לבצע שינויים.	ניתן להשתמש במופעים של Wholesaler באופן זהה למופעים של Person, מכיוון ש-Wholesaler יורשת מ-Person ומורחבת את הפונקציונליות שלה מבלי לשנות את אופן הפעולה שלה. משמעות הדבר היא שבכל מקום שמצפים ל-Person, ניתן להשתמש ב-Wholesaler ללא בעיות.	המחלקה מספקת ממשק קוהרנטי על ידי התמקדות בייצוג סיטונאים. היא לא חושפת שיטות או התנהגויות מיותרות, ושומרת על הממשק שלה נקי ותמציתי.
Invoice	למחלקת Invoice יש אחריות יחידה - ליצור חשבונית בהתבסס על הפורמט שסופק ומידע ההזמנה. היא מקפסלת את הלוגיקה לעיצוב והצגת פרטי החשבונית.	ניתן להרחיב אותה כדי לתמוך בפורמטים נוספים של חשבוניות מבלי לשנות את הקוד הקיים שלה. לדוגמה, אם יוכנס פורמט חדש, כמו "OnlineInvoice", ניתן להוסיפו מבלי לשנות את ההתנהגות הקיימת.	ניתן להשתמש במופעים של Invoice באופן זהה בכל מקום שמצפים ל-Invoice. התנהגות המחלקה מוגדרת היטב ועקבית עם ייעודה, ומאפשרת שימוש פולימורפי בה.	המחלקה מספקת ממשק ממוקד עבור יצירת חשבוניות בהתבסס על פורמטים שונים. היא לא חושפת שיטות או התנהגויות מיותרות מעבר למה שנדרש לאחריות שלה.
DeliveryCompany	למחלקת DeliveryCompany יש אחריות יחידה - לייצג חברת שליחויות ולספק פונקציונליות הקשורה לתמחור משלוחים. היא מקפסלת את הלוגיקה לניהול איש קשר ולחישוב מחירי משלוח.	המחלקה פתוחה להרחבה אך סגורה לשינוי. היא משיגה זאת על ידי מתן שיטה מופשטת getPriceForDelivery, המאפשרת לתת-מחלקות ליישם לוגיקת תמחור משלהן מבלי לשנות את המחלקה DeliveryCompany עצמה.	ניתן להחליף תת-מחלקות של DeliveryCompany במופעים של המחלקה הבסיסית בכל מקום שמצפים ל-DeliveryCompany. כל תת-מחלקה צריכה לכבד את החוזה המוגדר על ידי השיטות המופשטות, ולהבטיח שימוש התחלפי בהן.	המחלקה מספקת ממשק קוהרנטי ליצירת אינטראקציה עם חברות שליחויות. הוא כולל שיטות לגישה לאיש קשר, חישוב מחירי משלוח וקבלת שם החברה. שיטות אלה רלוונטיות למושג של חברת שליחויות ואינן חושפות התנהגות מיותרת.
DHL_Company	למחלקת DHL_Company יש אחריות יחידה - לייצג את חברת השליחויות DHL ולספק פונקציונליות הקשורה לחישוב מחירי משלוח. היא מקפסלת את הלוגיקה לקביעת מחיר משלוח של מוצר בהתאם לסוג המשלוח (אקספרס או רגיל).	המחלקה פתוחה להרחבה אך סגורה לשינוי. היא יורשת מהמחלקה DeliveryCompany ומספקת מימוש משלה לשיטה getPriceForDeliveringAProduct כדי לחשב מחירי משלוח ספציפיים ל-DHL. זה מאפשר הרחבות עתידיות על ידי הוספת תת-מחלקות חדשות עבור חברות שליחויות אחרות מבלי לשנות את הקוד הקיים.	ניתן להשתמש במופעים של DHL_Company בכל מקום שמצפים ל-DeliveryCompany. היא מכבדת את החוזה המוגדר על ידי המחלקה האב שלה על ידי יישום השיטות המופשטות הנדרשות (getPriceForDelivery, getPriceForDeliveringAProduct, getName) ומתן פונקציונליות ספציפית ל-DHL.	המחלקה מספקת ממשק ממוקד ליצירת אינטראקציה עם שירותי המשלוחים של DHL. היא כוללת שיטות לחישוב מחירי משלוח ולקבלת שם החברה, הרלוונטיות לפונקציונליות של DHL. היא לא חושפת התנהגות מיותרת.
FedEx_Company	למחלקת FedEx_Company יש אחריות יחידה - לייצג את חברת השליחויות FedEx ולספק פונקציונליות הקשורה לחישוב מחירי משלוח. היא מקפסלת את הלוגיקה לקביעת מחיר משלוח של מוצר בהתאם לסוג המשלוח (אקספרס או רגיל).	בדומה ל-DHL_Company, מחלקה זו יורשת מהמחלקה DeliveryCompany ומספקת מימוש משלה לשיטה getPriceForDeliveringAProduct. תכנון זה מאפשר הרחבת ההתנהגות מבלי לשנות את הקוד הקיים, לפי העיקרון של פתיחות להרחבה אך סגירות לשינוי.	ניתן להשתמש במופעים של FedEx_Company באופן זהה למופעים של המחלקה האב DeliveryCompany. היא תואמת את החוזה המוגדר על ידי המחלקה האב על ידי יישום השיטות הנדרשות (getPriceForDelivery, getPriceForDeliveringAProduct, getName) ומתן פונקציונליות ספציפית ל-FedEx.	המחלקה מספקת ממשק ספציפי ליצירת אינטראקציה עם שירותי המשלוחים של FedEx, ומציעה שיטות לחישוב מחירי משלוח ולקבלת שם החברה. היא לא חושפת התנהגות מיותרת, ועוקבת אחר העיקרון של הפרדת ממשקים בהתבסס על צרכי המשתמש.

Dependency Inversion Principle
המחלקה מתקשרת עם מחלקות אחרות רק דרך הפונקציות של המחלקות Person Product ובכך ממשת את עיקרון זה.
המחלקה לא מממשת אף ממשק, אבל היא מספקת קבוצת שיטות קוהרנטיות שקשורות לניהול נתוני מוצר והזמנות. לקוחות יכולים ליצור אינטראקציה עם שיטות אלה בהתאם לצרכים שלהם מבלי להיות תלויים בפונקציונליות מיותרת.
למחלקה אין תלות במחלקות אחרות ובכך קל יותר לתחזק אותה
מסתמכת על ההורשה מ Product, עבור הפונקציונליות הבסיסית שלה, מה שמוביל לגמישות וביתוק תלות ממימושים קונקרטיים

מסתמכת על ההפשטה שמספקת Customer ועוקבת אחר העיקרון של תלות, Person מחלקת בהפשטות ולא במימושים קונקרטיים. זה מקדם גמישות וניתוק תלות בבסיס הקוד
מסתמך על ההפשטה שמספקת Wholesaler ועוקבת אחר העיקרון של תלות, Person מחלקת בהפשטות ולא במימושים קונקרטיים. זה מקדם גמישות וניתוק תלות בבסיס הקוד
היא מסתמכת על ההפשטות Order ו- Person כדי ליצור חשבוניות. זה מקדם גמישות וניתוק תלות בבסיס הקוד.
המחלקה מסתמכת על הפשטות ולא על מימושים קונקרטיים. היא מסתמכת על ההפשטות Order ו- Person כדי לחשב מחירי משלוח ולנהל פרטי קשר. זה מקדם גמישות וניתוק תלות, ומאפשר שימוש חליפי במימושים שונים של Order ו- Person.
המחלקה מסתמכת על הפשטות (DeliveryCompany, Order, Product), (IsExpress, Person) ולא על מימושים קונקרטיים. זה מקדם גמישות וניתוק תלות, ומאפשר שימוש חליפי במימושים שונים של הפשטות אלה.
בדומה ל- DHL_Company, מחלקה זו מסתמכת על הפשטות ולא על מימושים קונקרטיים, ומקדמת גמישות וניתוק תלות. היא מסתמכת על ההפשטות המסופקות על ידי המחלקה הבסיסית DeliveryCompany ועל מחלקות קשורות אחרות (Order, Product, IsExpress, Person).