המכללה הטכנולוגית כנרת
ע"ש אלפרין בעמק הירדן בע"מ - חל"צ
הנדסאי - תואר שהוא גם מקצוע

הגדרות התרגיל

במסגרת התרגיל, תתבנתו מערכת שתייצג חנות נעליים. יהיה עליכם לנהל את הסחורה בחנות במבנה נתונים של 'מערך דינאמי' של נעליים, ולאפשר ממשק נוח לעבודה מול החנות.

במסגרת התרגיל, תממשו שלוש מחלקות: המחלקה PairOfShoes שתייצג זוג נעליים, ShoeStorage ש'יאחסן' את הנעליים בחנות, והמחלקה ShoeStore שתייצג את חנות הנעליים.

על המימוש שלכם להיות ברור ונקי, ולעלות בקנה אחד עם עקרונות ה-OOP שנלמדו בקורס עד כה.

המחלקה PairOfShoes

על מחלקה זו לייצג זוג נעליים. לזוג נעליים יש שם – מחרוזת שתייצג את שם הנעל, ומחיר – ערך ממשי המייצג את עלות הזוג.

על המחלקה לתמוך בבמה שיטות (methods):

- בנאי (Constructor) – בהינתן שם נעל ומחיר נעל ה-constructor יאתחל את שדות האובייקט.
- GetName – שיטה זו תחזיר את שם הנעל.
- GetPrice – שיטה זו תחזיר את מחיר הנעל.
- SetPrice – שיטה זו תשנה את המחיר של הנעל.

שימו לב שאין צורך ב-Setter לשדה השם של הנעל, כיוון שלא נרצה לאפשר למתכנת לשנות את שמות הנעליים.

המחלקה ShoeStorage

המחלקה ShoeStorage תנהל 'מחסן' של הנעליים שבחנות.

'מחסן' זה יממומש בצורה של "מערך דינאמי" של נעליים (הסבר בהמשך). המחלקה ShoeStorage תכיל מערך של מצביעים לזוגות נעליים (PairOfShoes); כאשר מערך זה יתמלא בנעליים, מערך הנעליים יועתק למערך אחר, הגדול פי 2 ממנו, באופן שקוף למשתמש. כך, למעשה, יהיה ניתן להוסיף מספר בלתי מוגבל של נעליים למחסן (עד אשר ייגמר הזיכרון במחשב).

על מחלקה זו להכיל שני שדות:

- מצביע לתחילתו של מערך מוקצה דינאמית של מצביעים ל-PairOfShoes
- אורכו של מערך זה (מספר שלם).

על מחלקה זו לתמוך בבמה שיטות:

- בנאי (Constructor) – בהינתן גודלו ההתחלתי של ה-ShoeStorage, הוא יאותחל, כך שתחילה כל המצביעים לנעליים במערך יאותחלו ל-NULL. גודל ברירת המחדל של מחסן הנעליים יהיה 4.
- AddPairOfShoes – בהינתן נעל (PairOfShoes), הנעל תתווסף למאגר הנעליים של החנות על ידי הוספת הנעל למערך הנעליים. במידה ומערך הנעליים מלא, על ה-method לשכפל את מאגר הנעליים לתוך מערך בגודל כפול, ולהוסיף את הנעל. זיכרו – על ה-entries במערך שלא מצביעים לנעל להצביע ל-NULL.
- RemovePairOfShoes – בהינתן שם של נעל, על שיטה זו להסיר את אחד העותקים של הנעליים בחנות עם אותו השם. במידה ולא קיימות נעליים בחנות עם השם שהועבר, שיטה זו לא תעשה דבר.
- GetPrice – בהינתן שם של נעל, שיטה זו תחזיר את מחירה. במידה והנעל לא קיימת במאגר, על השיטה להחזיר מחיר 0.0. ניתן להניח שכל הנעליים עם אותו שם הינן בעלות מחיר זהה.
- AverageShoePrice – שיטה זו תחזיר את המחיר הממוצע של כלל הנעליים שבמחסן (שימו לב – על כל הנעליים להיכלל בממוצע, בין אם יש כמה עם אותו שם ובין אם לאו). במידה ואין נעליים במחסן, יש להחזיר 0.0.
- Copy Constructor, Assignment Operator, Destructor – מאחר שאובייקט זה מנהל זיכרון דינאמי, יש לממש את ה-Big Three עבורו.

המחלקה ShoeStore

מחלקה זו תייצג את חנות נעליים. חנות נעליים תכיל מאגר של נעליים (ShoeStorage) וערך ממשי שייצג את אחוז ההנחה של על הנעליים בחנות זו (אחוז ההנחה יהיה ערך בין 0.0 ל-100.0).

על המחלקה לתמוך בשיטות הבאות:

- בנאי (Constructor) - בהינתן אחוז ההנחה על הנעליים בחנות, יש לאתחל את חנות הנעליים. אחוז ההנחה ברירת המחדל הינו 0%.
- getter, setter - SetDiscountPercent, GetDiscountPercent לערך אחוז ההנחה בחנות.
- AddShoes – בהינתן שם של נעל העלות שלה וכמות הזוגות, יתווספו למחסן הנעליים נעליים בהתאם לכמות הזוגות שהועבר למתודה. ערך ברירת המחדל של כמות הנעליים שמתווספות הינו 1. מחיר הנעל שמועבר לשיטה זו הינו מחיר לפני הנחה.
- AverageShoePrice – מחיר ממוצע של נעל בחנות (בהתאם לכמות הנעליים בחנות). מחיר ממוצע זה יהיה לאחר הנחה. במידה ואין נעליים בחנות, יוחזר מחיר ממוצע של 0.0.
- GetShoePrice – בהינתן שם של נעל, שיטה זו תחזיר את המחיר של הנעל לאחר הנחה. ניתן להניח שלכל הנעליים בעלות אותו שם במחסן, יש אותו מחיר.
- RemoveOnePair – בהינתן שם של נעל, אחת מזוגות הנעליים עם שם זה תוסר מהמחסן. במידה ואין נעל עם אותו שם – שיטה זו לא תעשה דבר.

<u>הערה</u>: ישנן כמה דרכים לתכנת מחלקות אלה כך שיתמכו בממשק הנ"ל, חלקן יעילות יותר, חלקן יעילות פחות. תוכלו לתכנת בכל דרך העולה על רוחכם, כל עוד היא עולה בקנה אחד עם עקרונות ה-OOP, ומבלי לשנות את חתימות המתודות הנ"ל.