**תרגיל** 2 - **Inheritance**

**תרגיל 1 - נא לקרוא את כל התרגיל לפני התחלת הכתיבה**

* **בכל פרויקט הנכתב במטודולוגיה מונחת עצמים יש לזהות בשלב הראשון של ניתוח המערכת את המחלקות המהוות את "טיפוסי הנתונים" המשתתפים בפרויקט, בתרגיל זה משתתפות המחלקות הבאות שכבר כתבתם:**

1. המחלקה **Poin**t המתארת נקודה במישור.
2. המחלקה **Rectang**le המכילה כשדה אובייקט מסוג **Point** , רוחב width- וגובהה המלבן- height.
3. מכוון שתתבקשו להוסיף מחלקה Circle אשר בינה לבין המלבן תכונות משותפות, במטרה להגדיר עיצוב נכון של התוכנה, יש לאוסיף מחלקת בסיס בשם **Shape** אבסטראקטית, שתכיל את כל המשותף לצורות הגרפיות הדו מימדיות שנגדיר בפרויקט, כלומר: position Point, צבע הצורה Color c ( לשם כך נא להשתמש במחלקה java.awt.Color; מתוך ה-**JCL**, ושדה אשר מסמן האם הצורה תצוייר עם מילוי או לא boolean fill, כמו כן יש להגדיר:

**ctors, get/set, draw(), remove() , move () ,resize(int percent)**. ) כרגע הפעולות

הגרפיות מדפיסות טקסט אל ה-Console, לכן לא ניתן לממש בצורה נכונה את remove ו-move).

1. יש להשתמש ככל האפשר במושגים:abstract, virtual, override,final . . .
2. יש לשנות את המחלקה Rectangle , כך שהיא כעת יורשת מ- Shape,כלומר כל מה שניתן, יש לעלות לרמת ה-Shape.
3. יש להוסיף מחלקה חדשה Circle היורשת מ-Shape המכילה בנוסף לשדות המתקבלים בירושה, שדה radius מסוג double , פונקציות לגישה set/get ,פונקציה draw ופונקציה לחישוב השטח, עיקף העיגול...
4. יש להגדיר מחלקה יורשת נוספת ל-Shape בשם Line המכילה נקודה נוספת , הקובעת ביחד עם נקודת מחלקת הבסיס, את התחלה וסוף של הקו. יש לממש גם כאן את הפונקציות **המתאימות** מתוך המלבן. (אין משמעות לשטח, עיקף.....)
5. יש ליצור ב- main מערך פולימורפי, המכיל 3 אובייקטים מסוגים שונים ולהפעיל עליהם את המטודות.

**תרגיל 2**

1. צור פרויקט חדש, אשר מכיל מחלקה בשם **Customer**, הכוללת: שם, כתובת, מספר לקוח סדרתי וייחודי, הנוצר אוטומטי עבור כל לקוח שנרשם , מחיר כולל לקניה לתשלום, תאריך רישום הלקוח מסוג Date, מחלקה שתכתבו בעצמכם ctors, get/set, toString().

למחלקה Customer יש להגדיר 2 מחלקות יורשות:

1. מחלקה **RegCustomer** המהווה לקוח רגיל, אשר נוסיף לו , תאריך יום הולדת גם מסוג Date.
2. מחלקה **VipCustomer** אשר תכיל מספר ייחודי נוסף, שזהו יהיה מספר חבר מועדון של הלקוח המועדף בחברה, כמו כן גובה ההנחה באחוזים עבור כל קניה שהלקוח מבצע.

בכל מחלקה ctor, get/set, פונקציה addToPayment( int amount) להוספת פריט, המקבלת

מחיר פריט נוסף המתווסף לעלות הקניה הכוללת,בכל אחד מסוגי הלקוחות לפי סוג החיוב. פונקציה

print() , ו- toString().

יש ליצור ב-main מערך של 3 אובייקטים מסוגים שונים ולהפעיל עליהם פולימורפיזם.

* **לאחר שהגדרנו את מחלקות סוגי הנתונים, יש להגדיר מחלקה נוספת אשר תכיל את מבנה הנתונים הפולימורפי שבפרויקט. כרגע נעשה זאת באמצעות מערך המקבל את גודלו ב-ctor.**

**תרגיל זה יאפשר לנו בהמשך לבצע refactoring כשנלמד Collections.**

**תרגיל 3**

1. בפרויקט של Shape הוסף מחלקה נוספת הנקראת ShapeContainer המכילה מערך של צורות גרפיות מסוג Shape, ומונה int count המסמן את הכמות הנוכחית של צורות בכל רגע נתון.
2. כמו כן בהיותו אובייקט המכיל אובייקטים אחרים, יש להגדיר את הפונקציות הבאות:

public ShapeContainer (int size) // constructor

public boolean add( Shape s)

public void drawAll()

public boolean remove( int index)

public boolean remove( Shape s)

public boolean removeAll()

public void resizeAll ( int percent )

1. השלב הבא, הינו להגדיר ב- main אובייקט אחד מסוג ShapeContainer, נמלא אותו בצורות שונות, ונפעיל עליהם פולימורפיזם.

**תרגיל 4**

**בפרויקט Customer –**

יש לחזור על אותם הסעיפים 10-12 , עבור שושלת המחלקות בפרויקט של Customer.

-----------------------------------------------------------------------------------------------------------------