

1 - משתני נגישות מגדירים את רמת הנגישות של השדה, מתודה או מחלקה עליה הם מצביעים - כלומר הם קובעים איפה בקוד נוכל להשתמש בהם. אנחנו נשתמש בהם כדי לממש כימוס, להסתיר הגיון פנימי של מחלקה ולחשוף את ההיגיון הרלוונטי למשתמש (או למחלקות אחרות). למשל במחלקה קואורדינטות, ניתן למשתמש להכניס X ו Y בזמן שבשדות הפרטיים של המחלקה נשמור את הנתונים בקוסינוס וטנגנס.

2 - זאת מילה שמורה שמשנה את הפונקציונליות של השדה עליה היא מצביע, כמו למשל override שמסמן את זה שאנחנו נותנים מימוש חדש למתודה שירשנו או access modifiers שמשנים את הרשאות הגישה של מה שהם מצביעים עליו (public, protected internal, internal, private protected, or private).

3 - עבור מחלקות - internal, עבור שדות - private

4 - במקרה שלנו כרגע זה כאשר שני (או כל מספר אחר) של פרוייקטים מצביעים אחד על השני, אנחנו יכולים לפטור את זה בכך שרק אחד מהפרויקטים יכיל את השני בפרופרטיז.

5 - הם נועדו להביע לנו שליטה על הקלט שאנחנו מקבלים להגדרת השדות של המחלקה. למשל במחלקת כספומט לא נאפשר הפקדה של סכום שלילי

```
Public void setDeposit(double deposit){  
  
    if(deposit <= 0){  
        Throw new Exception("Cant deposit 0 or less");  
    }else{  
        _deposit = deposit;  
    }  
  
}
```