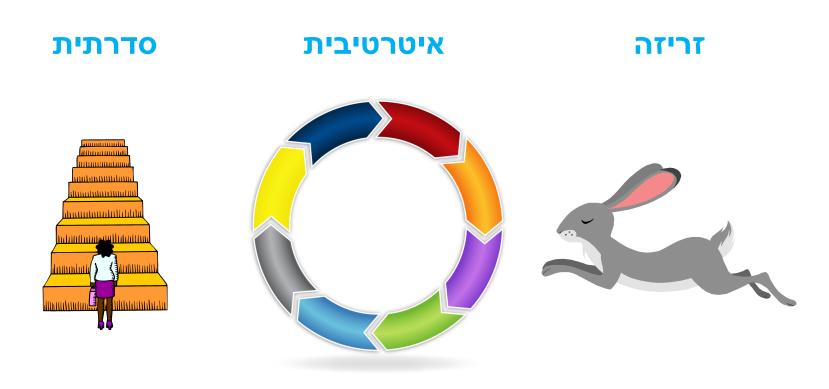


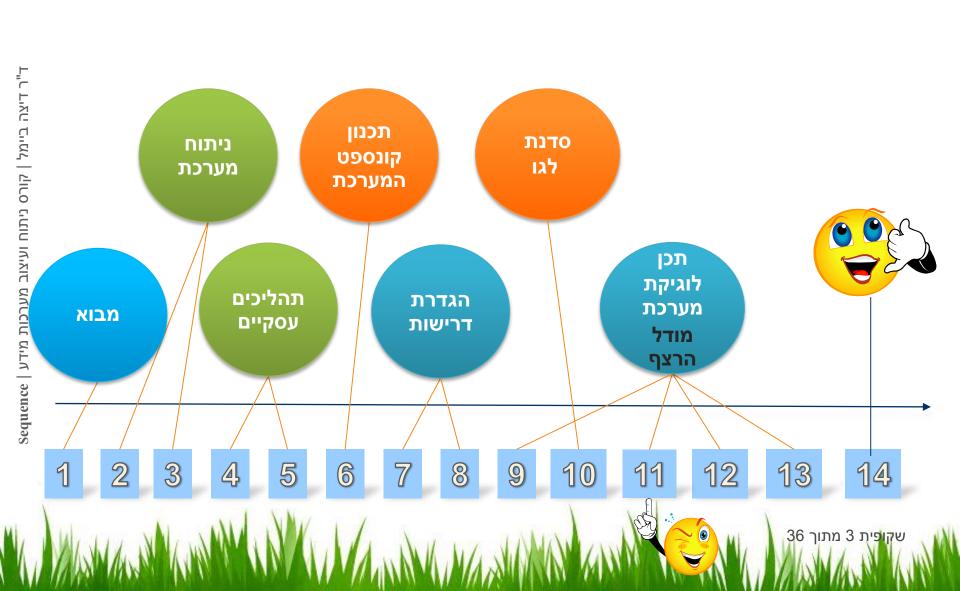
ניתוח ועיצוב מערכות מידע הרצאה 8: מודל הרצף - Sequence Model

ערבה צורי aravabt@gmail.com

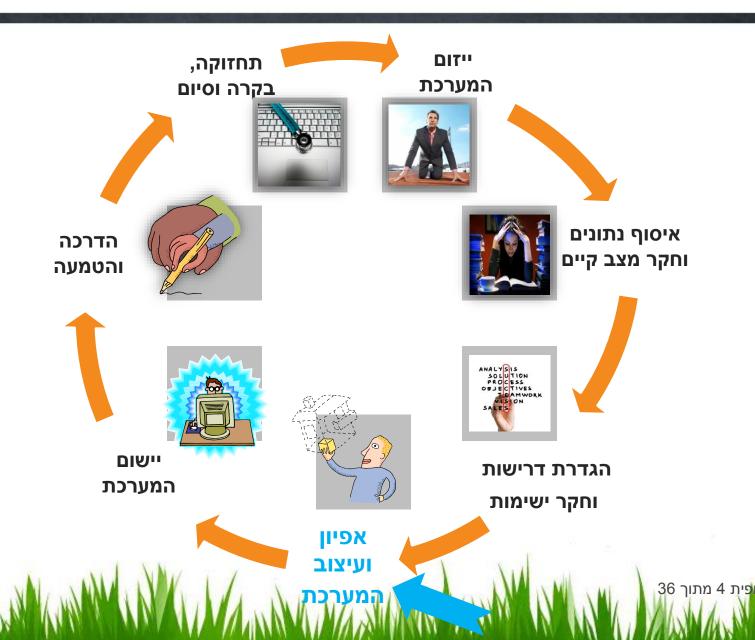
# סיכום השיעור הקודם: מתודולוגיות פיתוח



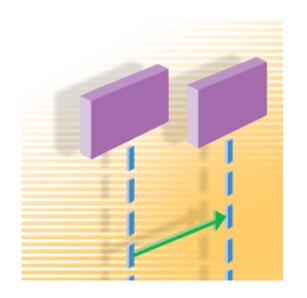
# סילבוס על פני ציר הזמן



#### מחזור החיים של מערכת המידע



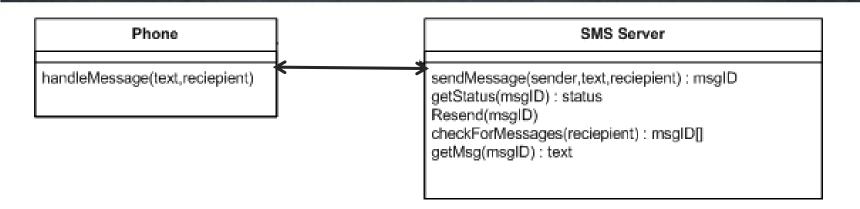
- הקדמה
- סימונים בסיסיים
- מסלולים חלופיים
  - מודולאריות •



## המודל הדינאמי (התנהגותי)

- המודל הדינאמי מתאר היבטים דינמיים של המערכת קרי התנהגות האובייקטים לאורך ציר הזמן
  - ניתן לתאר את התנהגות האובייקטים מזוויות ראיה שונות:
    - האינטראקציה בין האובייקטים המשתתפים בתרחיש כלשהו
      - מחזור החיים של האובייקט
      - − בקרת הזרימה של הפעילויות השונות במערכת
      - עבור כל זווית ראיה נעשה שימוש בתרשים שונה.

#### מדוע למדל התנהגות?



- מודל המחלקות מציג את המחלקות ואת הקשרים ביניהן, אך אינו מתאר כיצד מופעי המחלקה (האובייקטים) מתנהלים לאורך ציר הזמן.
  - למשל, מהו **הסדר** של הפעלת הפונקציות?
  - ? sendMessage, getStatus, resend
  - ? getStatus, sendMessage, checkForMessages
    - ?resend למשל, מתי אנחנו משתמשים בפונקציית –

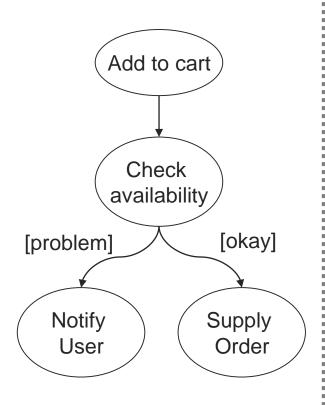
#### סוגי הדיאגרמות הדינמיות

- את סדר ותזמון Sequence diagram .1 העברת המסרים בין האובייקטים בתרחיש כלשהו העברת המסרים בין האובייקטים בתרחיש כלשהו
- כ collaboration diagram .2 **תרשים שיתוף** מציג את הארגון collaboration diagram .2 המבני של האובייקטים המעבירים את המסרים.
- state diagram .3 **תרשים מצבים** מציג את השינויים החלים בחיי האובייקט בעקבות אירועים במערכת.
  - activity diagram .4 **תרשים פעילויות** מציג את בקרת הזרימה activity diagram .4 בין הפעילויות, קצת מזכיר את תרשים הזרימה הפונקציונאלי.
    - .Use case אחרון חביב.

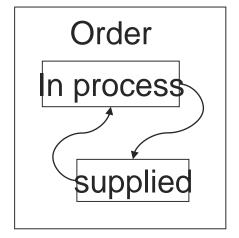
#### **Sequence Diagrams**

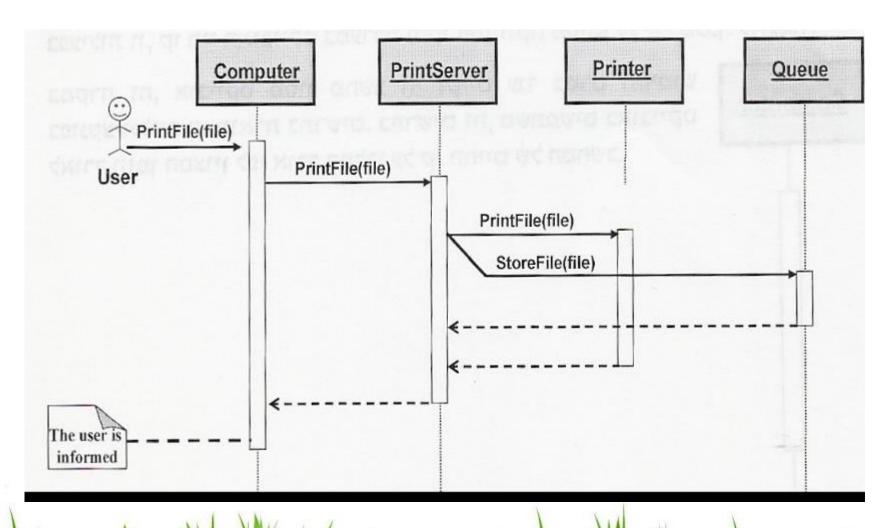
# ד"ר דיצה ביימל | Checkout Manager message: create | קורס ניתוח ועיצוב מערכות מידע | Sequence Order message: create Inventory

## **Activity Diagrams**

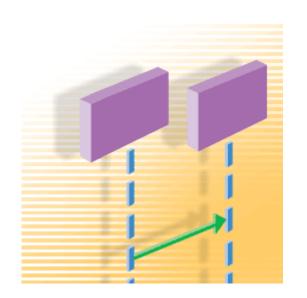


#### **State Diagrams**





- הקדמה
- סימונים בסיסיים
- מסלולים חלופיים
  - מודולאריות •



- ערשים רצף לוכד את ההתנהגות של use case מסוים תוך שימוש באובייקטים של המחלקות שהוגדרו בדיאגרמת המחלקות
  - מתוארת לאורך זמן באמצעות **מסרים use case התנהגות** של **messages)** המוחלפים בין אובייקטים על מנת להשיג מטרה
  - ✓ מסר הוא הפעלה של פעולה, או שליחת אות, או יצירה/הרס של אובייקטים
    - ✓ תרשים הרצף מתאר את המסרים המוחלפים בין האובייקטים✓ לאורך ציר הזמן.

therefore

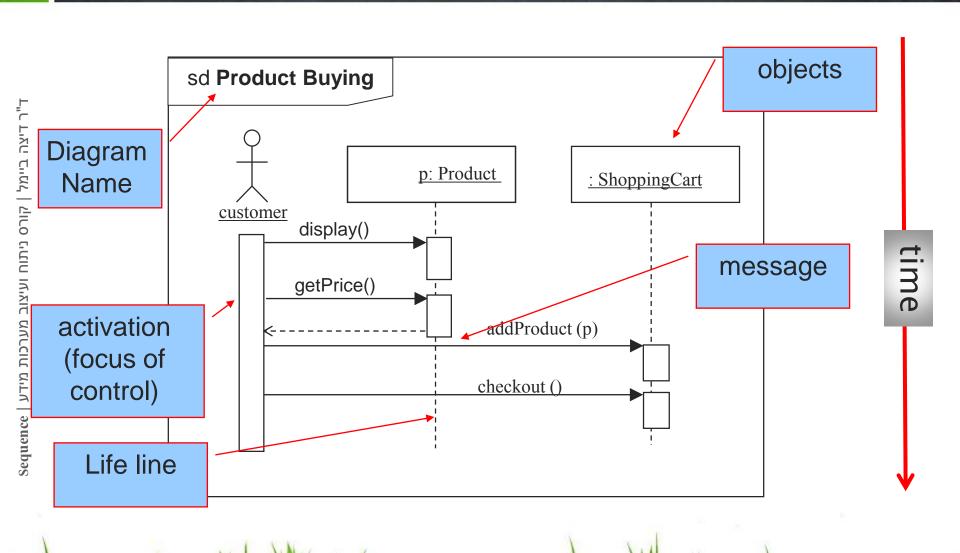
Sequence = Objects

+ messages

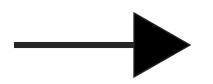
#### מאפייני תרשים הרצף

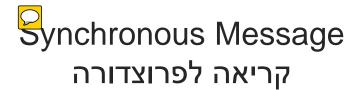
- ✓ תרשים הרצף מתאר את שיתוף הפעולה בין אובייקטים שונים במסגרת תרחיש כלשהו המשויך להתנהגות מסוימת.
- ערשים "ע"י תרשים ה-use case תתואר ע"י תרשים √ רצף נפרד אחד או יותר.
- ✓ האובייקטים המשתתפים בתרחיש, מוצבים בחלק העליון של הדיאגרמה, לאורך ציר ה X (מלבד אלו הנוצרים במהלך התרשים).
  - ✓ הדגש בתרשים זה הוא על רצף הזמן, אשר מפורט לאורך ציר ה- Y (מלמעלה למטה). כל מסר אשר נמצא מעל מסר אחר מציין פעולה מוקדמת יותר במימד הזמן.

# דוגמא לתרשים רצף



#### סוגים שונים של מסרים







synchronous Message שליחת סיגנל



Return Message מסר חזרה



משך הזמן בין שליחת מסר לקבלתו ע"י הְעצִם המתָאים:

זניח / לא זניח

.

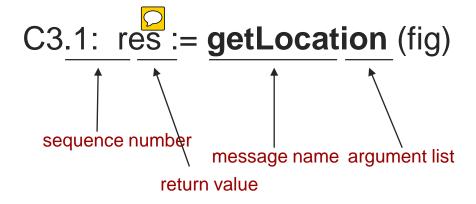


#### תנאים למשלוח מסר

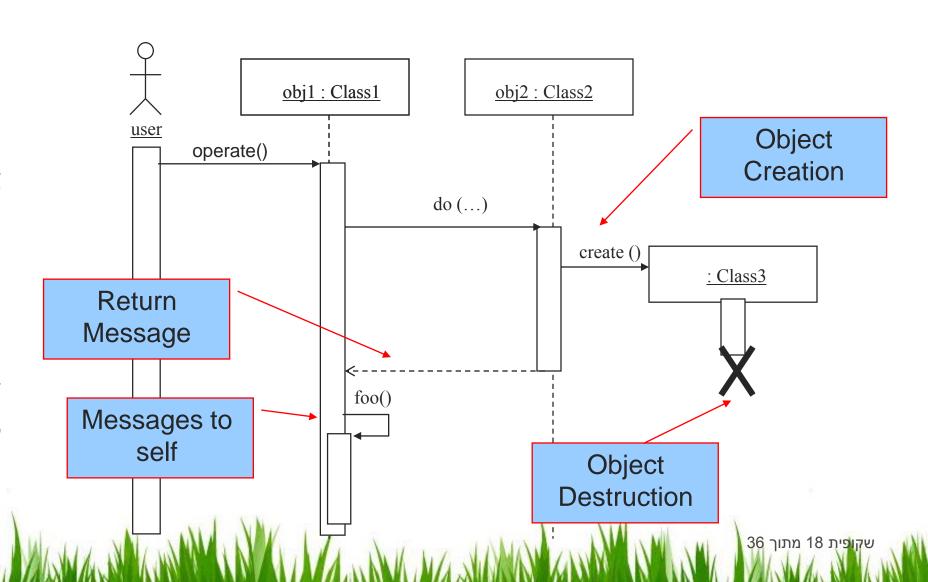
כדי שניתן יהיה לשלוח מסרים בין אובייקטים ממחלקות שונות חייבים להתקיים התנאים:

- ו. יש **קשר** בין המחלקות
- חייב להיות B לאובייקט A לאובייקט 10 חייב להיות 2. המסר (הסינכרוני) הנשלח מאובייקט 10 חלק מהשיטות של מחלקת אובייקט
  - 3. השיטה מוגדרת ומפורסמת **כציבורית**.

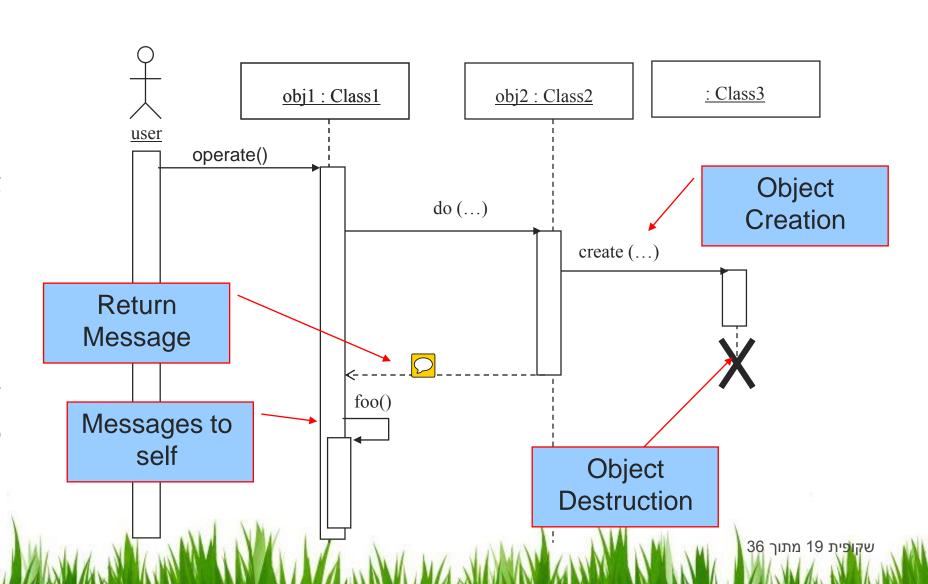
```
[sequence-number]
[return-value :=] [message-name] [(argument-list)]
```



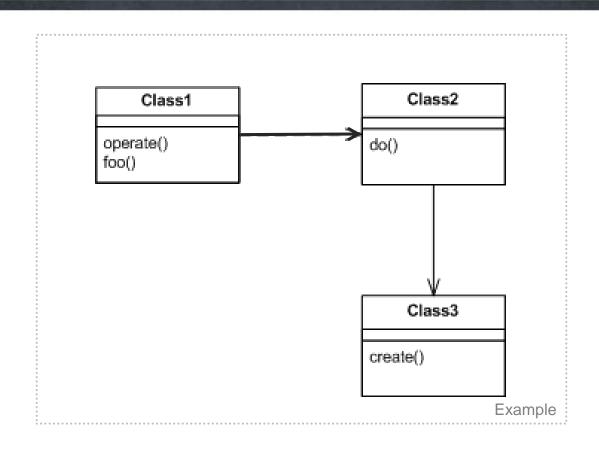
# דוגמא לתרשים רצף (1)



# דוגמא לתרשים רצף (2)

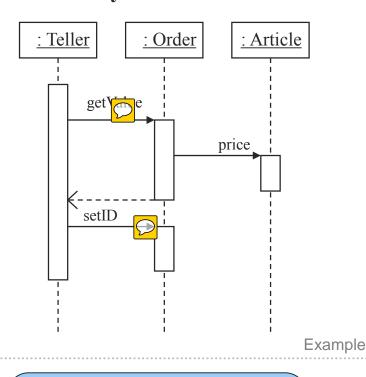


# דיאגרמת מחלקות תואמת

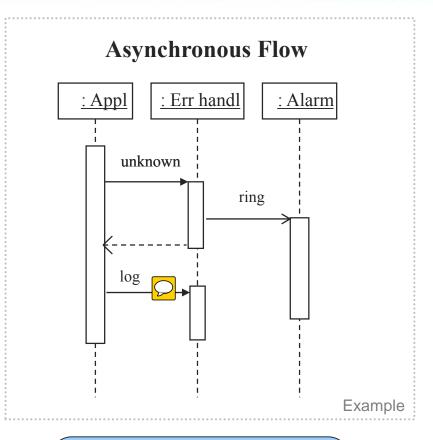


#### מסר סינכרוני ואסינכרוני

#### **Synchronous Flow**



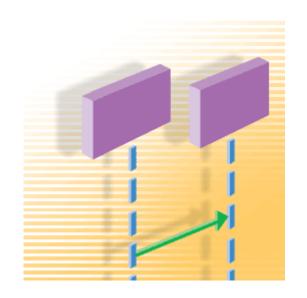
Price need to be finished, before teller can do another operation



Ring is executed, while the control flow is returned to Err\_ handle and Appl

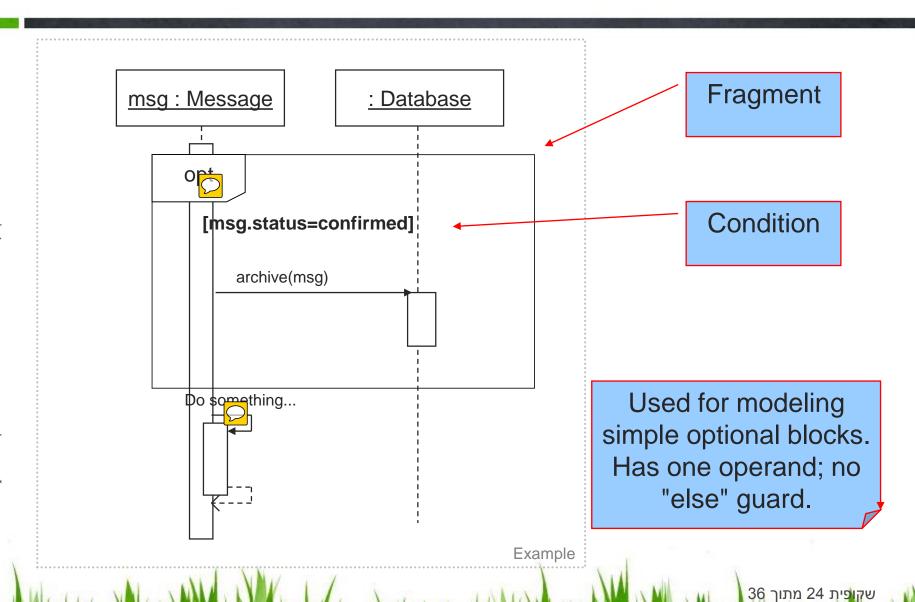
שקופית 🗠

- הקדמה
- סימונים בסיסיים
- מסלולים חלופיים
  - מודולאריות

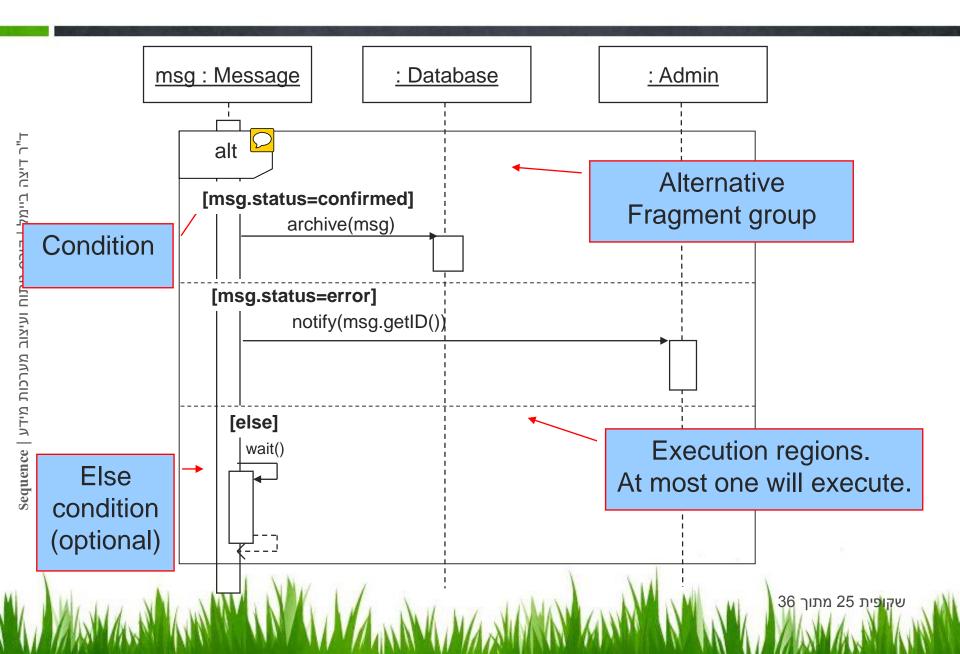


?כאשר אנחנו מתארים תרחיש - אילו מסלולים חלופיים קיימים

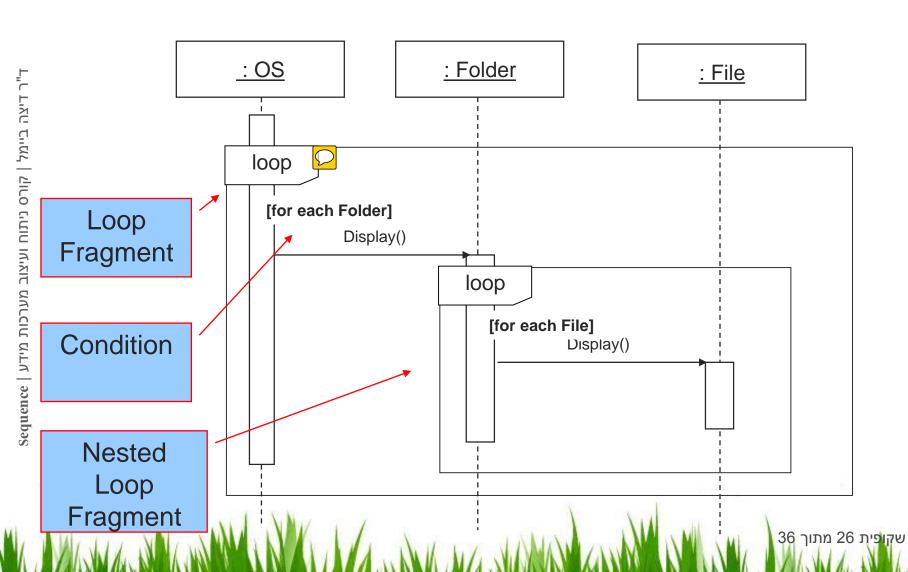
# **Options**

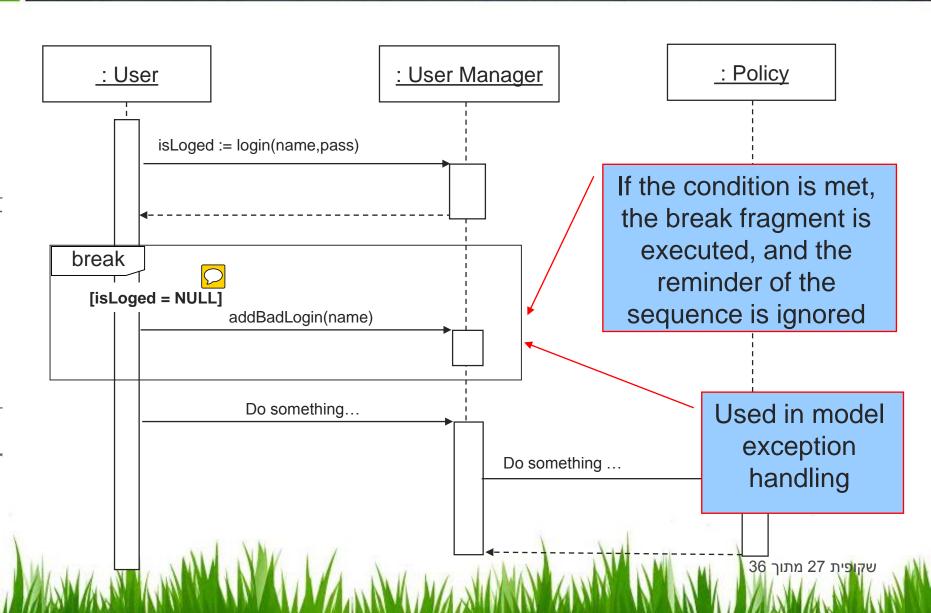


#### **Alternatives**



# Loops





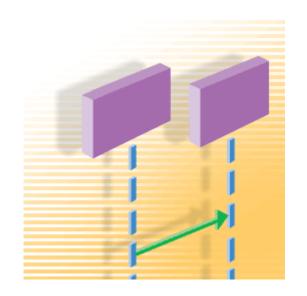
- [for each Object]
- [i=1..5]
- [status = okay]
- [status <> okay]
- [else]

**Usually for: Loops** 

Usually for: Option, Alternatives, break

No guard means an infinite loop

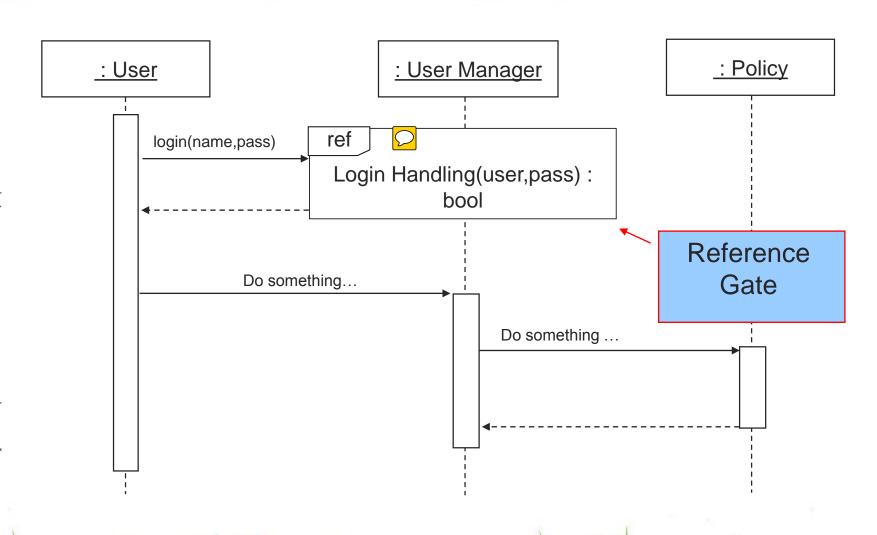
- הקדמה
- סימונים בסיסיים
- מסלולים חלופיים
  - מודולאריות



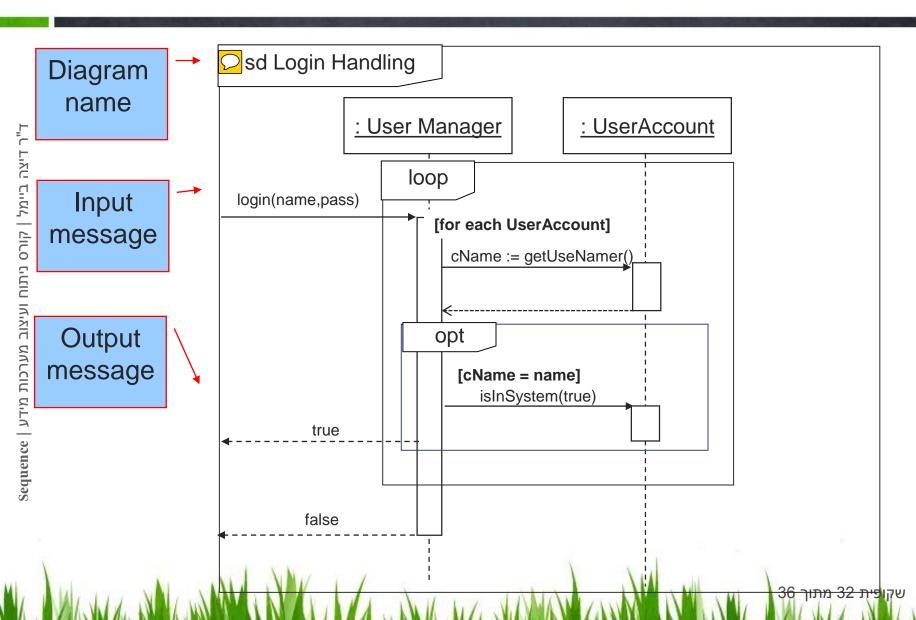


דרכים לייצר תרחישים מודולאריים

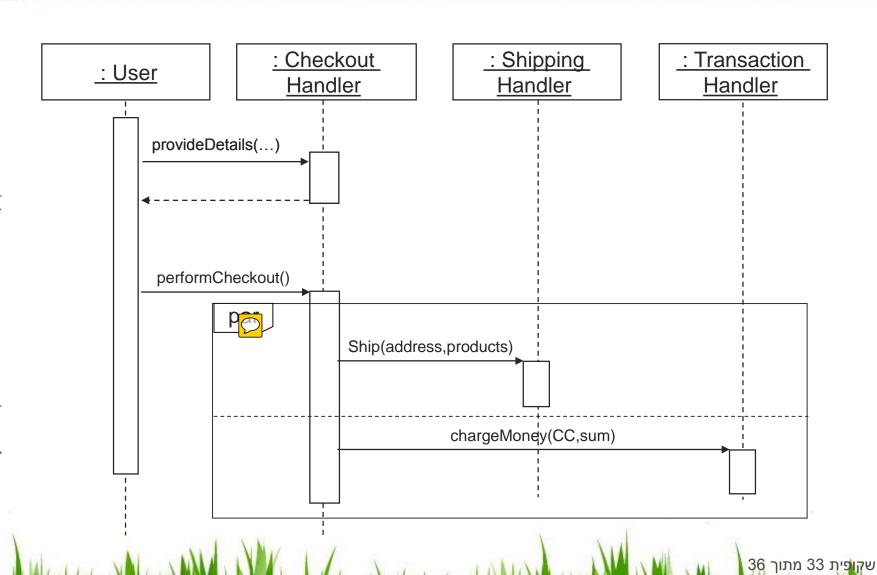
# Referencing a diagram



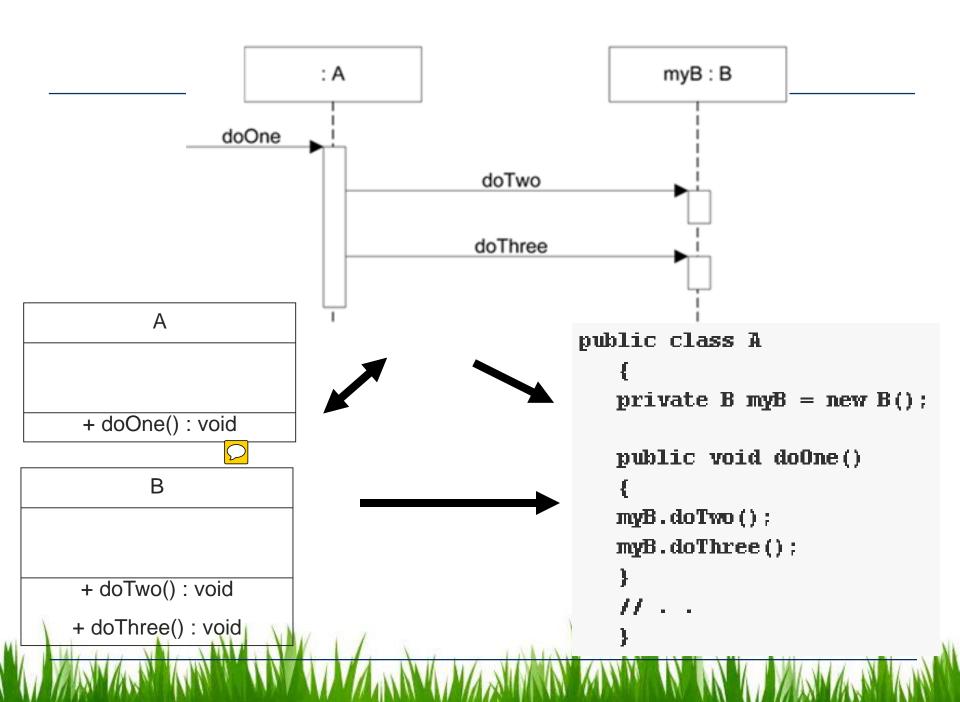
# **Referenced Diagram**



# **Parallel Fragments**



# מאבר מתראים ראף לקוד



- ע מידול דינמי (התנהגות ✓
- ערשים הרצף מורכב מאובייקטים ומסרים ✓
  - אלטרנטיביים למשל לולאות ✓
    - אריות למשל מקביליות ✓

