

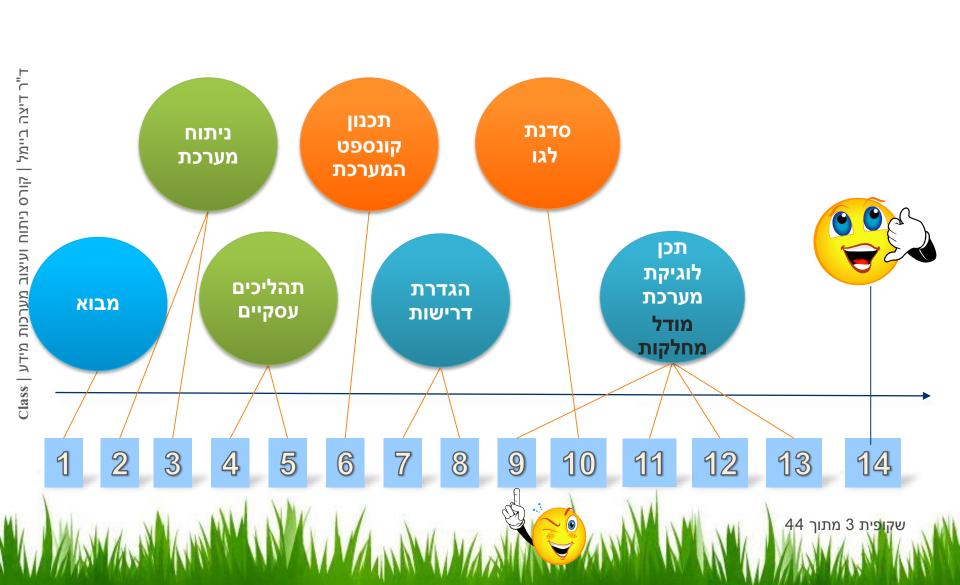
ניתוח ועיצוב מערכות מידע הרצאה 7: מודל המחלקות - Class Model

ערבה צורי aravabt@gmail.com

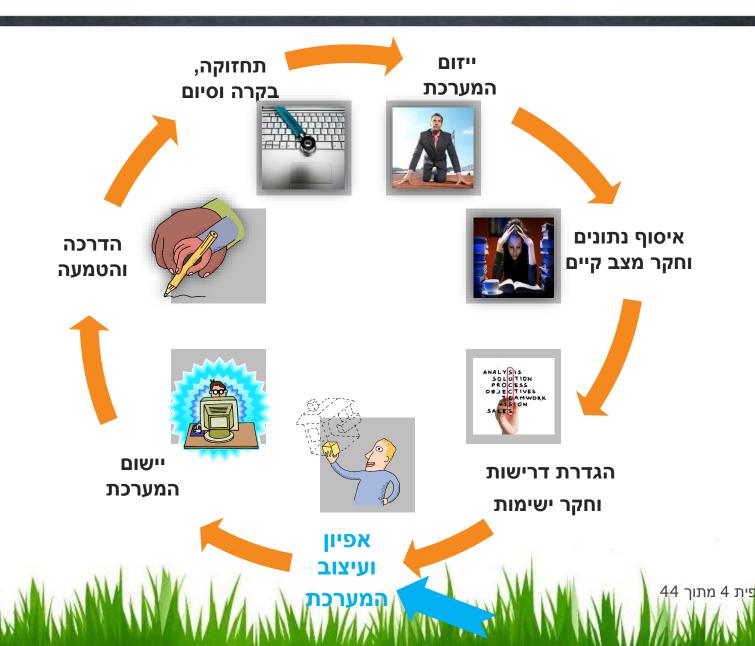
סיכום השיעור הקודם

- מכיל תרשים ומסמך Use Case ✓ מודל ה-
 - ✓ התרשים מבוסס כל שיטת סימון פשוטה
- . כל פונקציונאליות במערכת מבוטאת באמצעות בועה. ✓
- include, extend:הבועות יכולות לקיים ביניהן קשרים ✓
 - מפרט כל פונקציונאליות Use case ✓ מסמך ה-

סילבוס על פני ציר הזמן



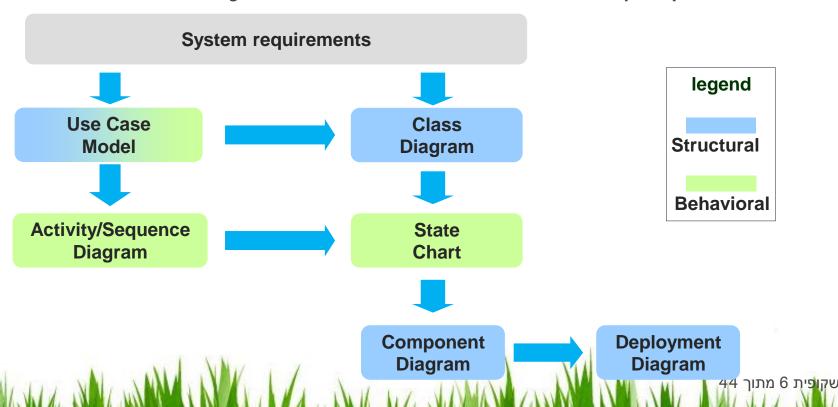
מחזור החיים של מערכת המידע



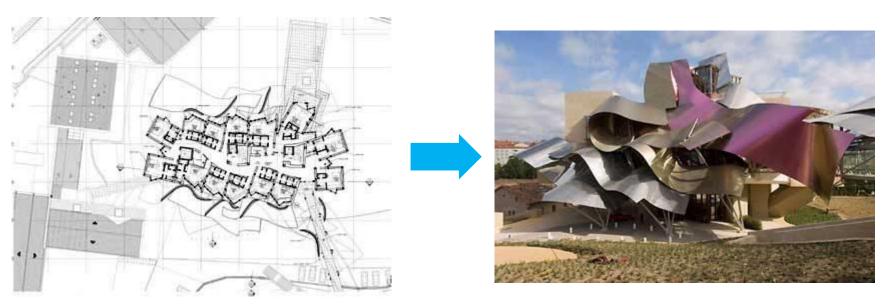
- הקדמה למודל המחלקות
 - מבנה המחלקה במודל
 - קשרים בין המחלקות
 - דיאגרמת אובייקטים
 - סיכום

מודל המחלקות Class model

- מודל זה הוא הלב של ה- UML
- המודל מתאר את מחלקות המערכת ואת הקשרים ביניהן
 - Object Oriented המודל תומך בקונספטים רבים של

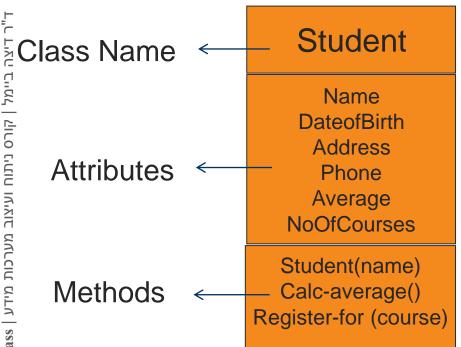


מהו מידול מבני? Structural Model



תכנון מבני מלבי את המאפיינים הקבוטים של התוצרים, כאלה שאינם מפתנים לאורך צמן

תזכורת: תכונות ושיטות של המחלקה



- (Attribute/Property) תכונות אוסף נתונים המתארים את המחלקה
- שיטות - (Service/Operation/Method) אוסף הפונקציות של המחלקה

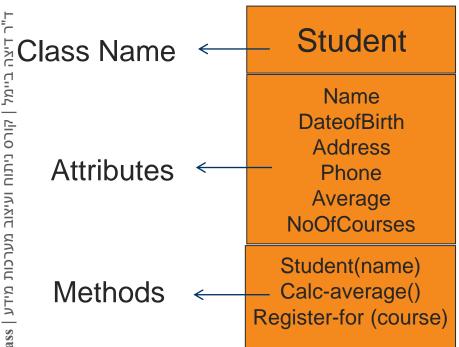
מידול מבני במערכות מידע

- מבנה סטטי של מודל מגדיר את:
- הישויות הקיימות (המחלקות, המנשקים, הקומפוננטות)
 - הקשרים בין הישויות
 - המבנה הפנימי של הישויות
 - מבנה סטטי של מודל **לא** מגדיר את:
 - ע המידע הטמפורלי (זמני −
 - התנהגות הישויות לאורך זמן
 - Runtime אילוצי –

תוכן ההרצאה

- הקדמה
- מבנה המחלקה בדיאגרמה
 - קשרים בין המחלקות
 - דיאגרמת אובייקטים
 - סיכום

תזכורת: תכונות ושיטות של המחלקה



- (Attribute/Property) תכונות אוסף נתונים המתארים את המחלקה
- שיטות - (Service/Operation/Method) אוסף הפונקציות של המחלקה

[visibility] name [[multiplicity]] [: type] [=initia value]



- ✓ visibility: the access rights to the attribute

 +name
- ✓ multiplicity: how many instances of the attribute middleName [0..1], phoneNumber [1..*]
- ✓ type: type of the attribute (integer, String, Person)
 salary: Real
- ✓ initial value: a default value of the attribute salary: Real = 10000, position: Point = (0,0)

```
+ isLightOn: booleaP = false
```

- numOfPeople: in

mySport

- + passengers[0..10]: Customer
- id: long

[visibility] name [(parameter-list)] [: return-type]

✓ A method can have 0 or more parameters, with syntax:

name: type [=default-value]

חתימת הפונקציה

מה ההבדל?

```
+ isLightOn(): boolean
```

- + addColor(newColor: Color)
- + addColor(newColor: Color): void
- # convertToPoint(x: int, y: int): Point
- changeItem(key: string, newItem: Item): int

תוכן ההרצאה



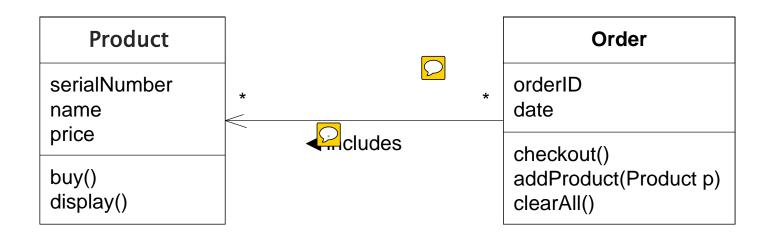
- הקדמה
- מבנה המחלקה במודל
- קשרים בין המחלקות
 - דיאגרמת אובייקטים
 - סיכום •

קשרים בין מחלקות

- (Association) זיקה
- היכרות" בין עצמים ממחלקות שונות
 - (aggregation) הכלה, הקבצה
- עצם ממחלקה כלשהי "מכיל" עצמים ממחלקה אחרת
 - (inheritance, generalization) הורשה, הכללה
 - הגדרת מחלקה על בסיס מחלקה אחרת

Association – קשר זיקה

Association קשר זיקה



- הקשר המבני **הפשוט** ביותר **והנפוץ** ביותר
- משמעות הקשר: אובייקטים משני צידי הקשר "מכירים" זה את זה
 - הקשר הוא **קבוע** לאורך זמן •

מאפייני קשר זיקה



• שם

(navigation) • •

 \square מהו כיוון הקשר \square

תפקיד י

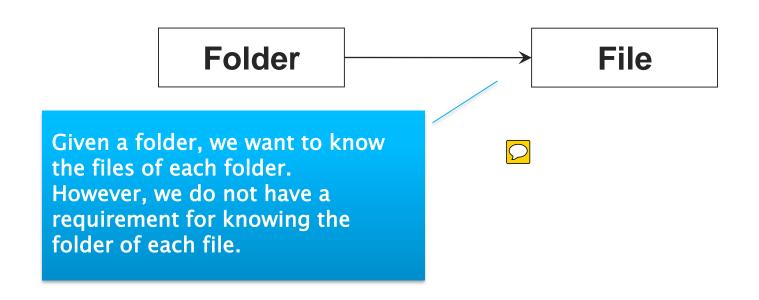
– מה התפקיד של כל אחד מהאובייקטים בקשר

- (multiplicity) •
- האם הזיקה קיימת תמיד
- האם ההכרות היא עם יותר מעצם אחד
 - *אילוצים

Multiplicity

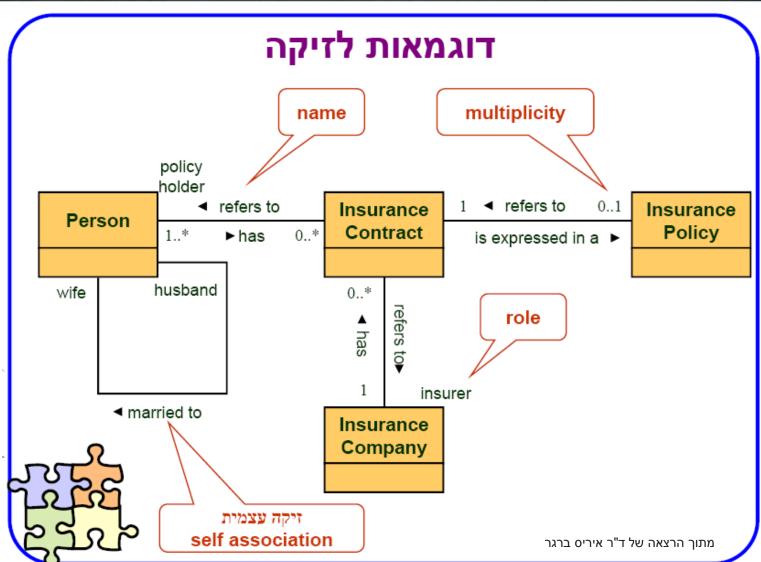
- ☐ 1:1 -default
- ☐ 3 exactly 3 object
- * (or n) unbounded
- \square 1..* 1 to eternity
- □ 3..9 3 to 9

Navigation כיוון הקשר

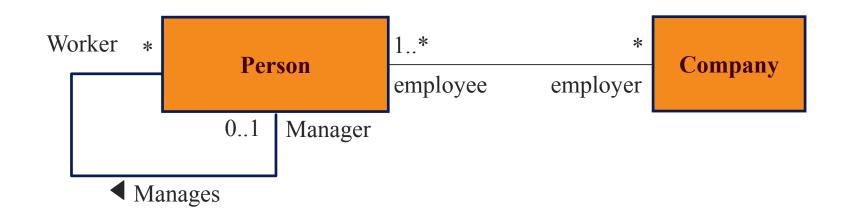


- י אם הקשר מכיל כיוון והוא **חד כיווני**, אזי "ההיכרות" היא חד כיוונית
- אם הקשר אינו מכיל כיוונים, אזי הוא קשר דו-כיווני, דהיינו, ההכרות היא הדדית
 - בד"כ יש להוסיף כיוון לקשר, אלא אם הדרישה היא לקשר דו-כיווני •

דוגמא לקשר זיקה

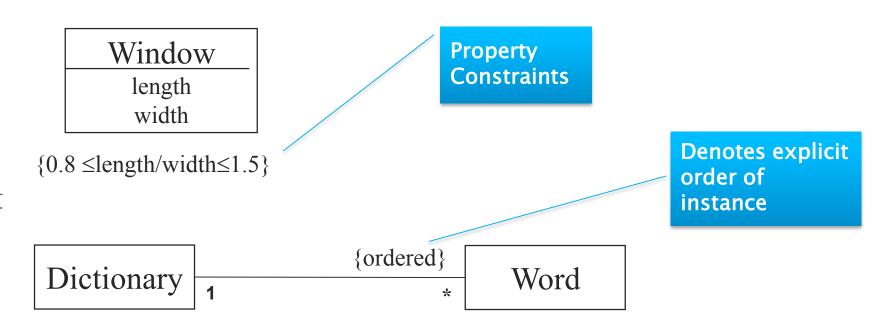


Roles תפקידים בקשר



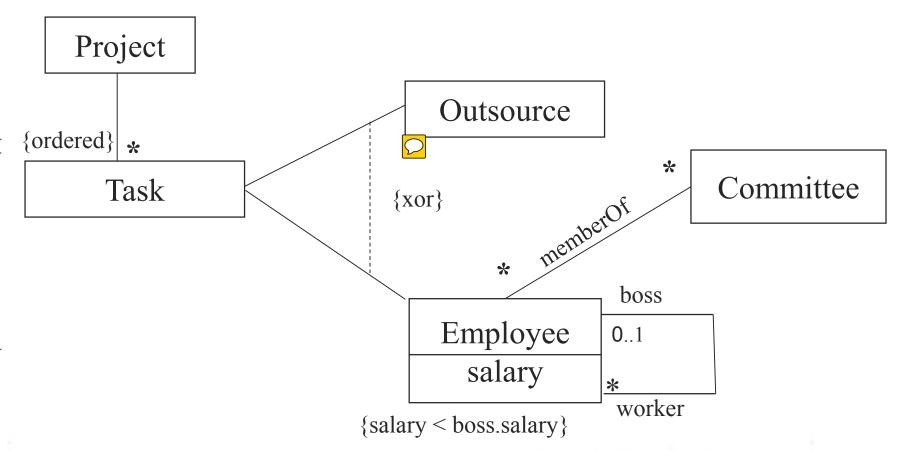
- ניתן להוסיף **שמות** של תפקידים בשני צידי הקשר
 - מוסיף הבנה למשמעות הקשר •
- self association בעיקר משמעותי בקשר זיקה עצמי

Constraints אילוצים

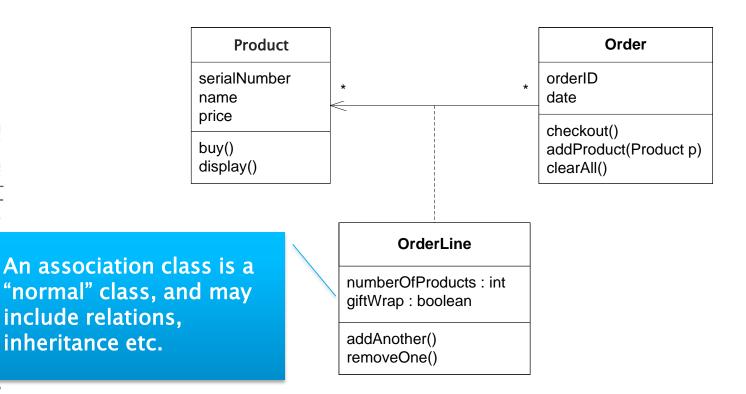


- רינו אילוץ סמנטי המצוין ע"י טקסט. האילוץ יכול להיכתב Constraint בצורה מתמטית פורמאלית, או בצורה לא פורמאלית
 - של UML השימוש באילוצים נעשה ברוב הדיאגרמות של
 - מסמנים אילוץ ע"י {} •

שקופית 25 מתוך 44



מחלקת זיקה Association class



- לפעמים לזיקה עצמה ניתן לתת מעמד של **מחלקה** •
- סימון מחלקת קשר ע"י סימן מקווקו לתוך אמצע קשר הזיקה •

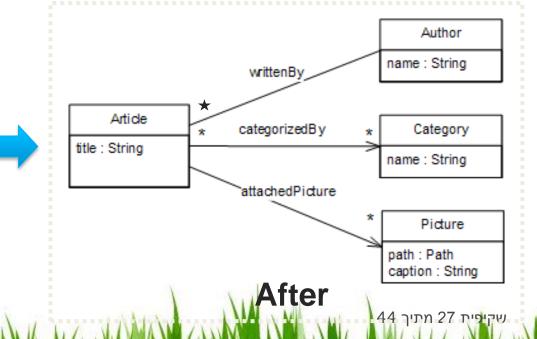
SSI VIEW CONCERN SINGLE

ד"ר דיצה ביימל | קורס

3

יש לנרמל תכונה במקרים הבאים:

- 1. תכונה צריכה להיבחר מתוך אוסף מוגדר וידוע מראש (למשל...)
 - 2. התכונה מייצגת מחלקה (למשל...)
- 3. חלק מהתכונות מקובצות לישות אחת בעלת התכונות (למשל...)



Article

title: String

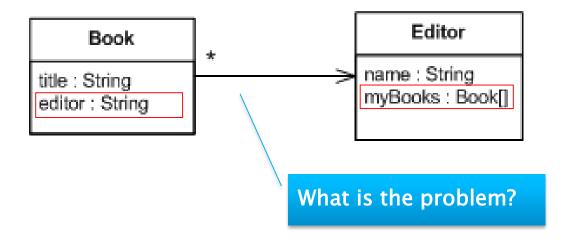
categories : String[]

author : String

attachedPictures: Path[] picturesCaption: String[]

Before

קשרים מול תכונות



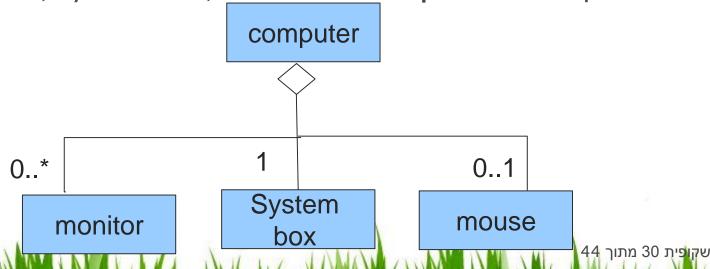
• מימוש הקשר מתבצע בשלב של יישום המערכת. קשר בין מחלקות יכול להתממש דרך מצביעים, מערכים, וכ"ו.

קשרי הכלה: Aggregation - קשר הקבצה Composition - קשר הרכבה

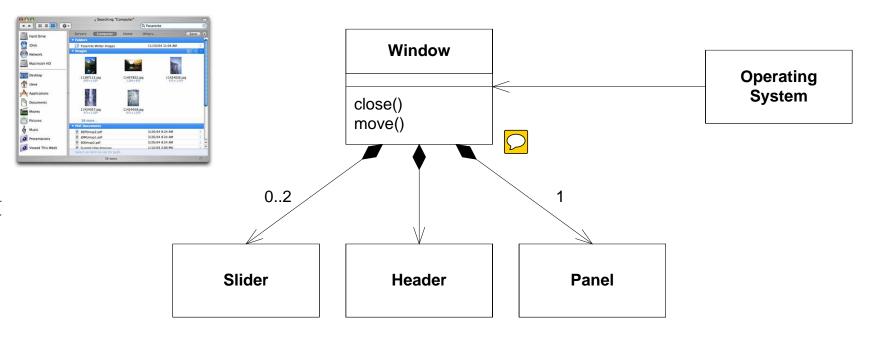
Aggregation קשר הקבצה

- הרחבה של קשר זיקה, קשר הכלה חלש
- מבטא את העובדה שמחלקה אחת היא חלק של מחלקה אחרת או
 מחלקות אחרות
 - כמו בקשר זיקה (multiplicity) פיתן להוסיף מאפיין של **ריבוי** •

"Monitor, System Box, and Mouse are part-of Computer"



קשר הרכבה Composition



- קשר חזק יותר מקשר הקבצה י
- מבטא את העובדה שמחלקה אחת היא הרכבה של מחלקה אחרת
 - האובייקט החלש חי בזכות האובייקט החזק
 - ביטול האובייקט החזק יבטל את האובייקט המוכל בו
 - ניתן להוסיף מאפיין של ריבוי (multiplicity) כמו בקשר זיקה •

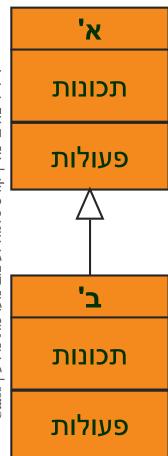
הקבצה מול הרכבה

Aggregation	Composition	
Part can be shared by several wholes category 04 * document	Part is always a part of a single whole Window * Frame	
Parts can live independently (i.e., whole cardinality can be 0*)	Parts exist only as part of the whole. When the whole is destroyed, they are destroyed	
Whole is not solely responsible for the object	Whole is responsible and should create/destroy the objects	

שקופית 32 מתוך 44

Inheritance – קשר הורשה

תזכורת: הורשה



- מחלקה ב' יורשת את מחלקה א':
- ב' מכילה את כל התכונות של א'
- ב' מכילה את כל הפעולות של א' –
- ב' מכילה את כל הקשרים של א' –
- בנוסף, ב' מכילה תכונות ופעולות משל עצמה
 - 'של א' (sub-class) ב' היא תת-מחלקה
- סימון ב-UML: חץ חלול המסתיים במחלקת העל.

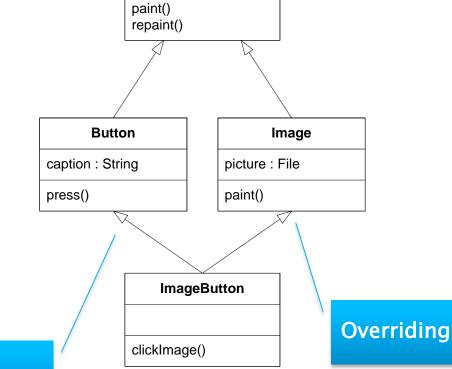
יתרונות ההורשה

אלימינציה לפרטיםמציאת אלמנטים משותפים

– הגדרת היררכיה

שימוש חוזר –





GraphicComponent

x:int y:int

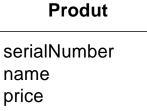
Multiple Inheritance

שקופית 35 מתוך 44

תוכן ההרצאה

- הקדמה
- מבנה המחלקה בדיאגרמה
 - קשרים בין המחלקות
 - דיאגרמת אובייקטים
 - סיכום

בדיאגרמת אובייקטים, אנו ממדלים מופעים של מחלקות •



buy() display() Apple Ipod: Product

In runtime

Sony Vaio: Product

name = "Vaio Portable" price = 2999 \$

serialNumber = 113234

Apple IMac: Product

name = "IMac 1C" price = 1456 \$

serialNumber = 184934

Class Diagram

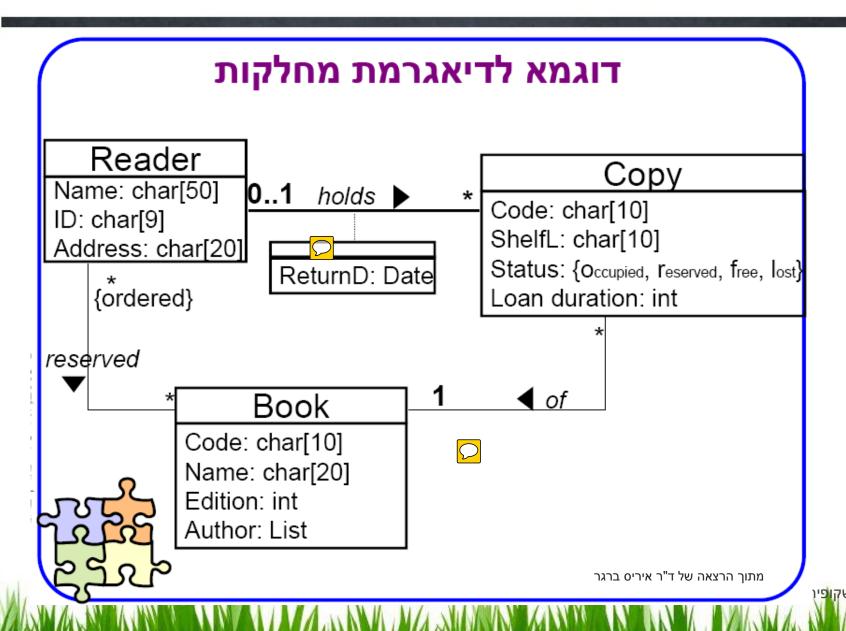
Object Diagram

שקו<mark>פ</mark>ית 37 מתוך 44

דיאגרמת אובייקטים

- דיאגרמות אובייקטים רלוונטיות לרגע מסוים בזמן ריצת המערכת •
- לכן, דיאגרמת אובייקטים "לוכדת" את המצב של האובייקטים, דהיינו, את אוסף הערכים שלהם
 - שיטות אינן כלולות בדיאגרמת אובייקטים •
 - אנחנו משתמשים בדיאגרמת אובייקטים:
 - למדל דוגמאות של המערכת בזמן ריצה
 - לזהות מצבים אסורים ומצבים חריגים

דוגמא לדיאגרמת מחלקות



מחלקות לעומת עצמים

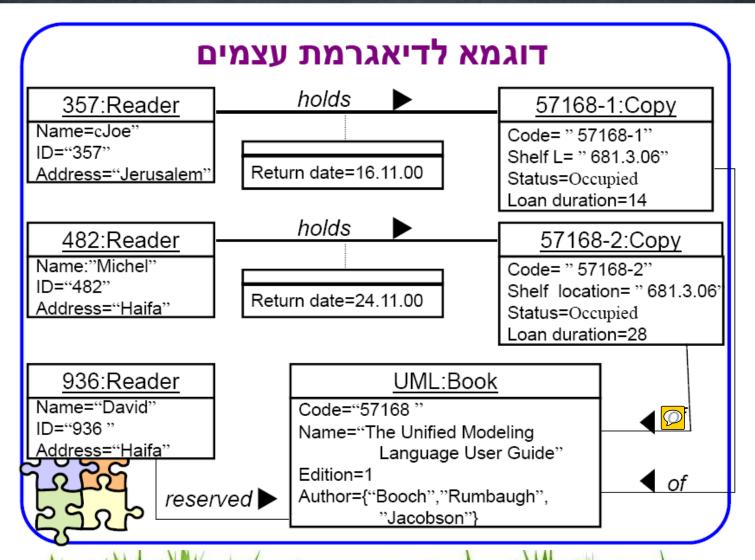
מחלקות לעומת עצמים

דיאגרמת עצמים	דיאגרמת מחלקות	
עצמים: שמות מופעים	זהות	מחלקות:
ערכים	תכונות	
	פעולות	
קשרים (links)	שם	זיקות:
	תפקיד (role)	(association)
	(multiplicity) ריבוי	
עצמי קשר (object link) עצמי קשר	(association class) מחלקות זיקה	

מתוך הרצאה של ד"ר איריס ברגר

שקופיו

דוגמא לדיאגרמת עצמים



תוכן ההרצאה

- הקדמה
- מבנה המחלקה בדיאגרמה
 - קשרים בין המחלקות
 - דיאגרמת אובייקטים
 - סיכום

- מידול מבני ✓
- ע קשרים בין מחלקות ✓
- Associations, constraints -
 - Compositions -
 - Generalization -

