

שיטות הנדסיות לפיתוח מערכות תוכנה

Class, Sequence and Activity Diagrams

כל תרשים בעל מטרה ייחודית משלו, לכן מתבצע בשלבים שונים של מידול המערכת

- ✓ מטרת activity היא לתאר זרימה (flow) של תהליכים במערכת, לכן מתבצע ראשון (לאחר הגדרת פונק' בסיסית באמצעות use case)
- ✓ בשלב הבא, נרצה לתאר את מחלקות המערכת והקשרים ביניהן באמצעות class diagram
- ✓ רק לאחר הגדרת המחלקות, נוכל לתאר את האינטראקציה בין האובייקטים השונים ואת סדר ותזמון ההודעות המועברות ביניהם באמצעות sequence diagram
- ✓ שימו לב: תהליך של יצירת תרשימי רצף הוא למעשה תהליך איטרטיבי בו משנים ומזקקים את תרשימי המחלקות



אורט בראודה
ORT Braude College

Activity Diagram

המטרה: לתאר זרימה (flow) של תהליכים

• מרכיבי דיאגרמה:

✓ צומת פעולה

• בד"כ, כניסה אחת ויציאה אחת

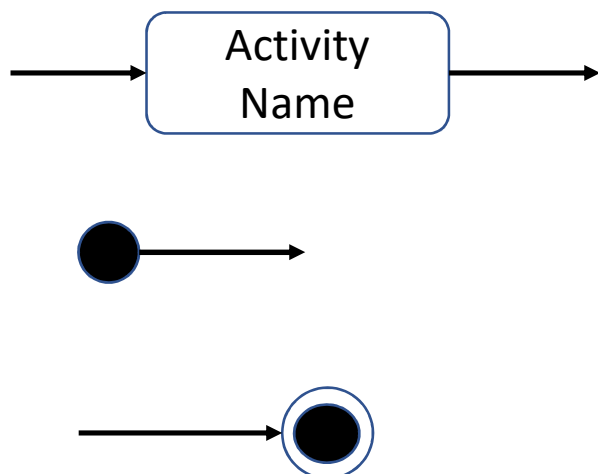
✓ התחלת פעילות

• יציאה אחת, ללא כניסה

✓ סיום פעילות

• כניסה אחת, ללא יציאה

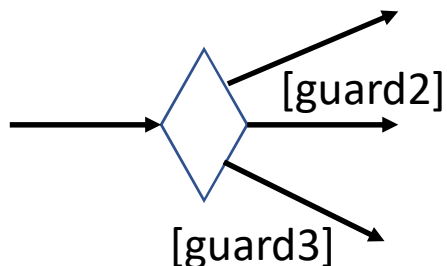
• עם הפעלתו נעצרת כל פעילות התרחיש



צמתי בקרה:

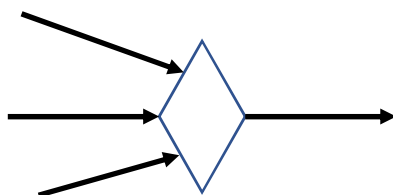
✓ החלטה (decision)

- כניסה אחת, יציאה מרובות
- סוג הפעילות ביציאה: XOR



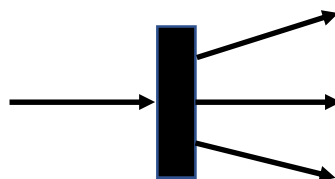
✓ התמזגות (merge)

- יציאה אחת, כניסות מרובות
- סוג הפעילות בכניסה: OR



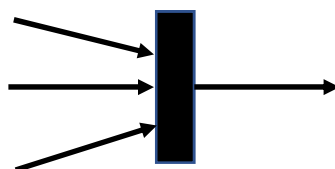
✓ מזלג (fork)

- כניסה אחת, יציאה מרובות
- סוג הפעילות ביציאה: AND

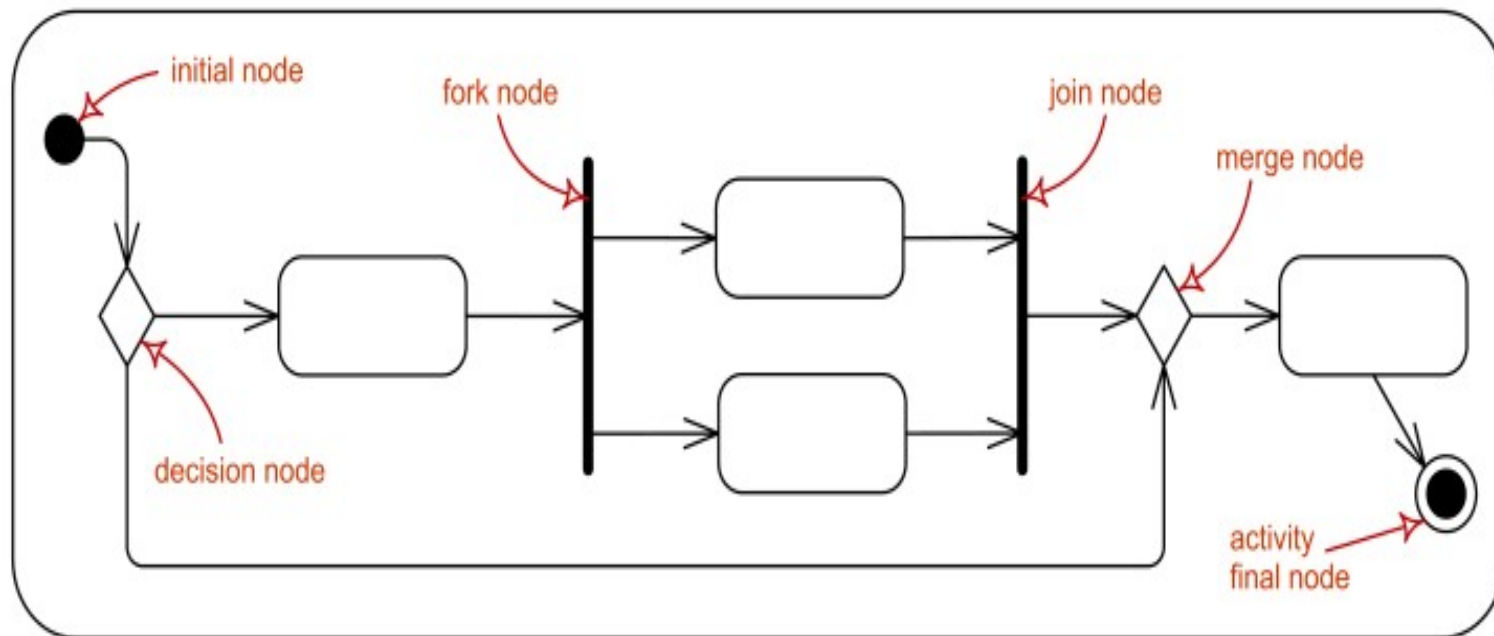


✓ הצטרפות (join)

- כניסה מרובות, יציאה אחת
- סוג הפעילות בכניסה: AND



דוגמא ביניים:



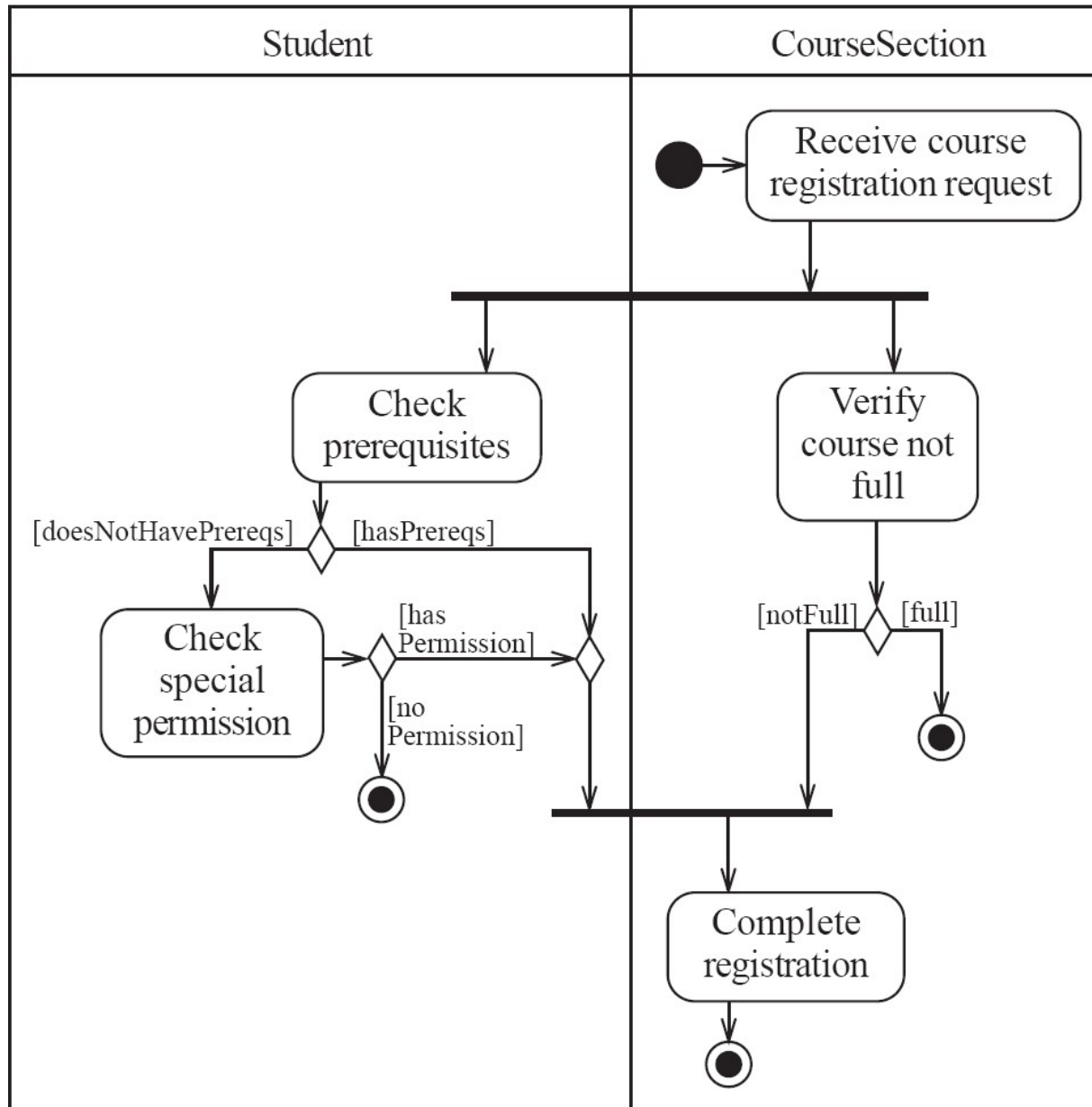
Partition/Swimlanes

- ניתן לחלק את הדיאגרמה למסלולים שונים

Partition	Partition3	Partition2

Partition4	
Partition5	
Partition6	

דוגמא מסכמת:





אורט בראודה
ORT Braude College

Class Diagram

המטרה: לתאר את מחלקות המערכת והקשרים ביניהן

• מחלקה מתוארת ע"י שלושת החלקים הבאים:

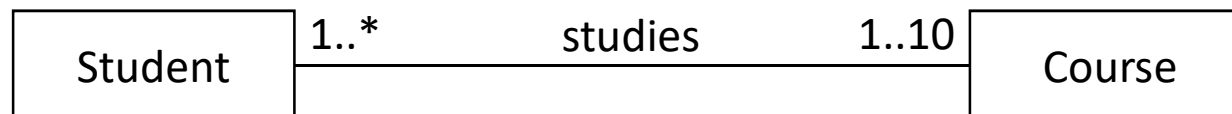
Class_Name
- attribute : type
+ operation() : type
+ operation(parameters) : type

• **Association** מתאר קשר זיקה ("היכרות") בין 2 מחלקות.

מתואר בדרך כלל ע"י פועל. יתכן גם קשר בין המחלקה לבין עצמה (self-association).

על כל קשר יכולים להופיע:

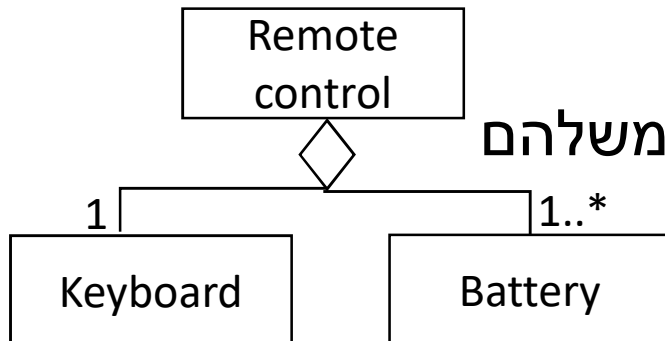
- **שם הקשר** (חד כיווני או דו כיווני).
- **ריבוי** (multiplicity) מציין כמה מופעים של המחלקה יכולים להיות מקושרים (link) למופע אחד של המחלקה השנייה. ברירת המחדל היא בדיוק 1.
- **תפקיד** (role) מציין מה תפקידו של כל אחד מהעצמים בקשר. שימוש ב-role נחוץ בעיקר כאשר הקשר מחבר בין המחלקה לעצמה (self-association) או כאשר קיימים מספר קשרים בין זוג מחלקות.



קשרים בין מחלקות:

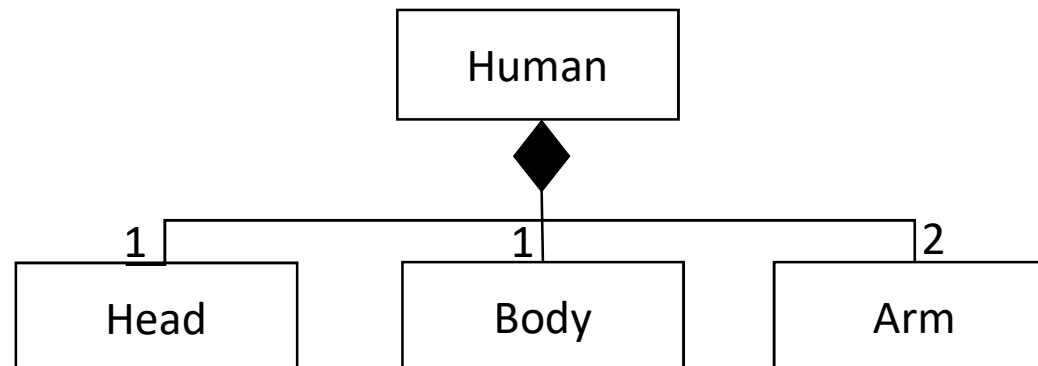
• קשר שלם וחלקיו

Aggregation ✓ - לחלקים יש קיום עצמאי משלהם



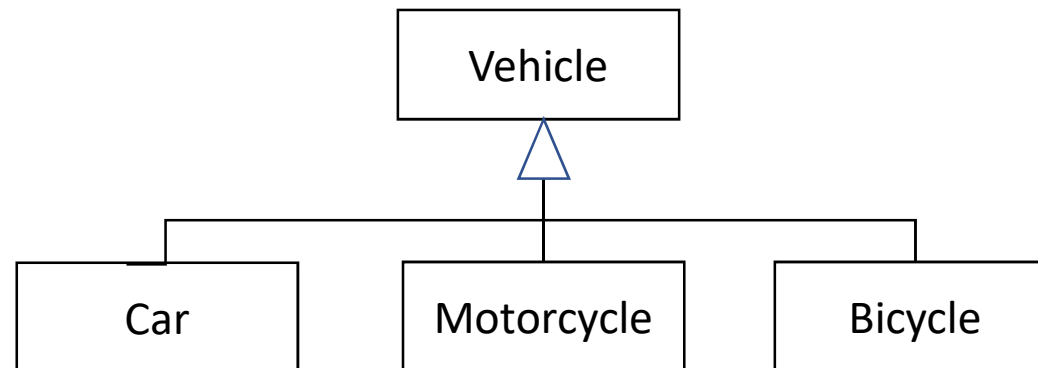
שימוש נוסף בקשר מסוג זה:
כאשר אדם מחלקה A יש מתודה המקבלת
כפרמטר אובייקט מטיפוס מחלקה B או מחזירה
אובייקט מטיפוס B.

Composition ✓ - קיומם של החלקים תלוי בקיומו של השלם



קשרים בין מחלקות:

- **הורשה, הכללה (inheritance, generalization) הגדרת מחלקה על בסיס מחלקה אחרת**
 - ✓ הרחבה של מחלקה קיימת
 - ✓ שינוי תצורה של מחלקה קיימת (פולימורפיזם)



מבחינים בשלושה סוגים של מחלקות, עפ"י תפקידם במודל המערכת:

✓ אובייקט **ממשק (boundary)**: מספק ממשק לישויות חיצוניות ("אקטורים", כולל תקשורת עם מערכות חיצוניות)



Boundary
object

✓ אובייקט **בקרה (control)**: שולט על התנהגות המערכת, אינטראקציות פנימיות, תזמון פעולות הכלולות בתרחיש



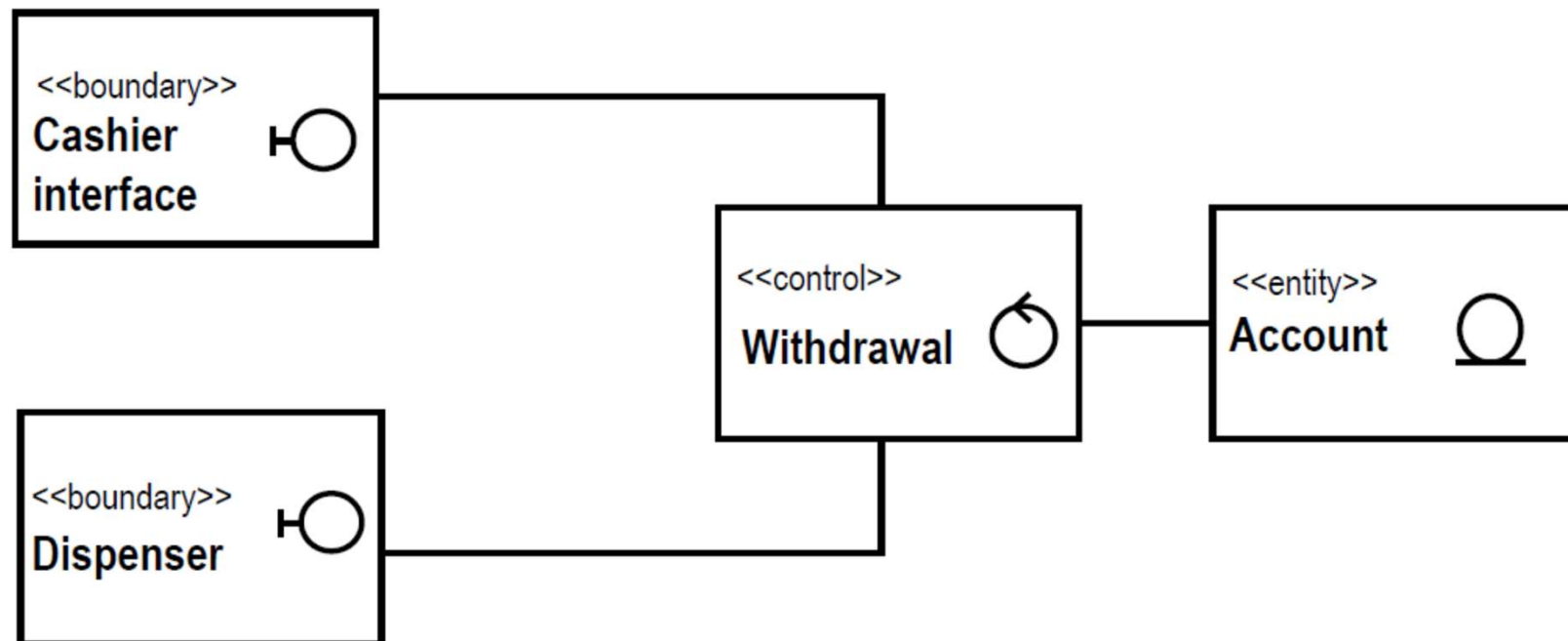
Control
object

✓ אובייקט **ישות (entity)**: בעיקר פסיבי, משמש לאחסנה קבועה של מידע לגבי תהליך, אירוע, או עצם מציאותי, כולל דרכי הגישה למידע



Entity
object

דוגמא:



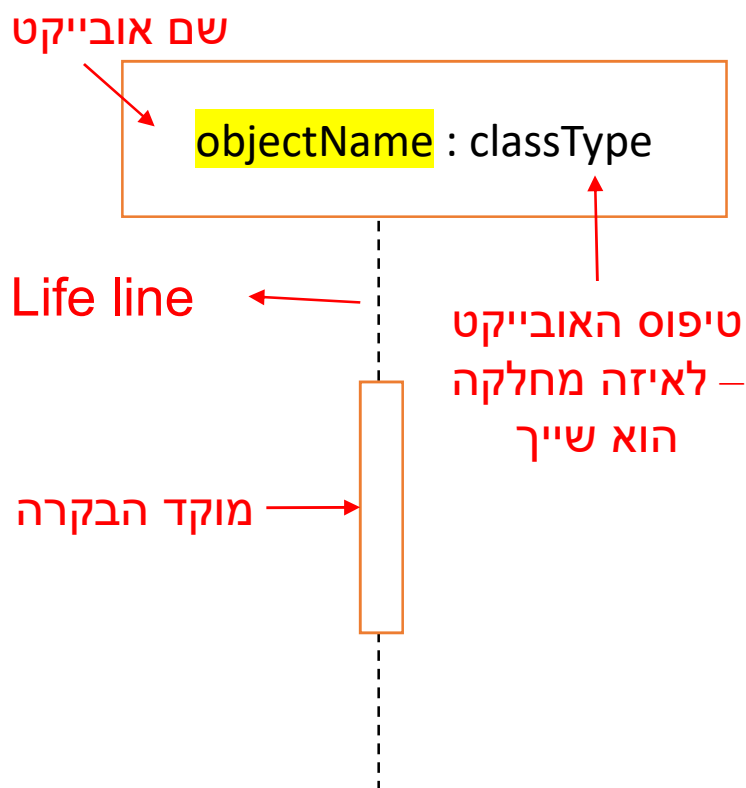


אורט בראודה
ORT Braude College

Sequence Diagram





המטרה: מתארת אובייקטים המשתתפים באינטראקציה ואת סדר ותזמון ההודעות המועברות ביניהם

- "קו החיים" (lifeline) של עצם הוא קו מקווקו המייצג את קיום העצם בתקופת זמן מסוימת.
- "מוקד הבקרה" (focus of control) הוא מלבן צר המתאר את פרק הזמן שבו העצם מבצע פעולות.



- מסר הינו הודעה המועברת בין עצמים
- במסר כלולים פרטי הודעה המועברת בין עצמים
- מסר מועבר בדרך כלל בין עצמים שיש ביניהם link

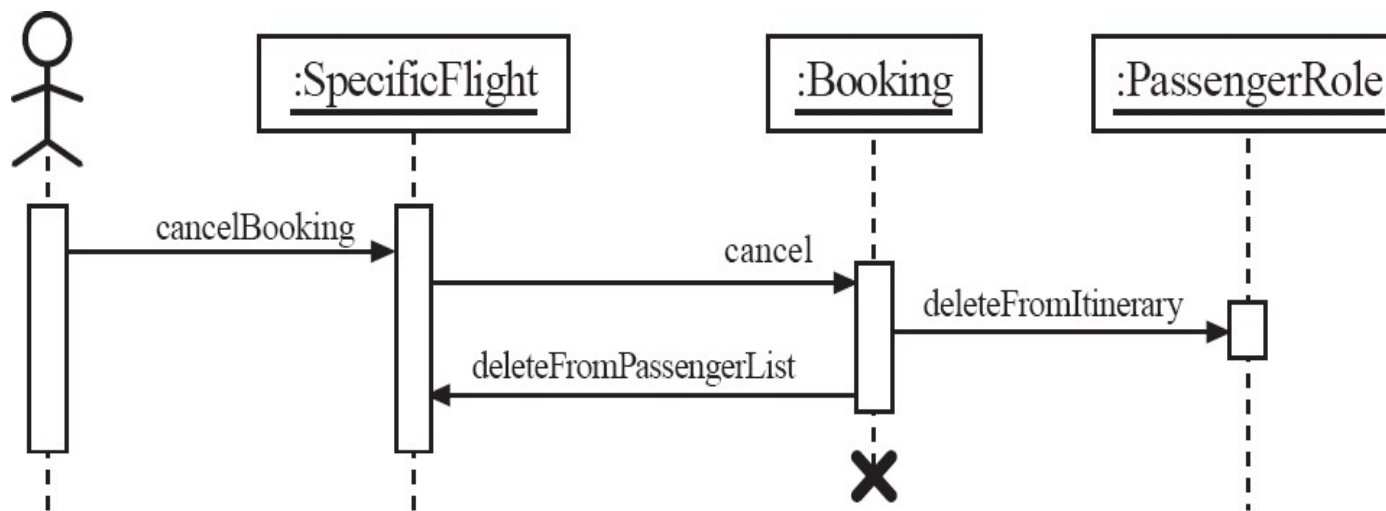
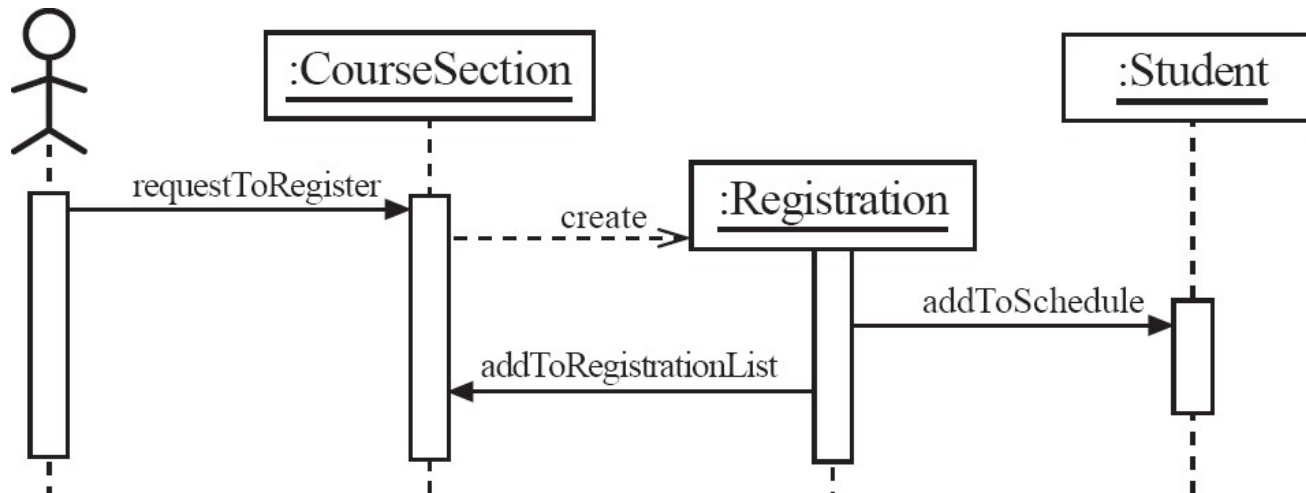
• סוגי מסרים:

- Call – קריאה לפרוצדורה
 • מעבר בקרה מקונן
- Send – שליחת סיגנל (א-סינכרוני)
 • שולח המסר ממשיך מיידית בשלב הבא של הביצוע לאחר שליחת המסר
- Return - מסר חזרה
 • בדרך"כ לא מופיע באופן מפורש
- Create - יצירה
 • יצירת עצם חדש

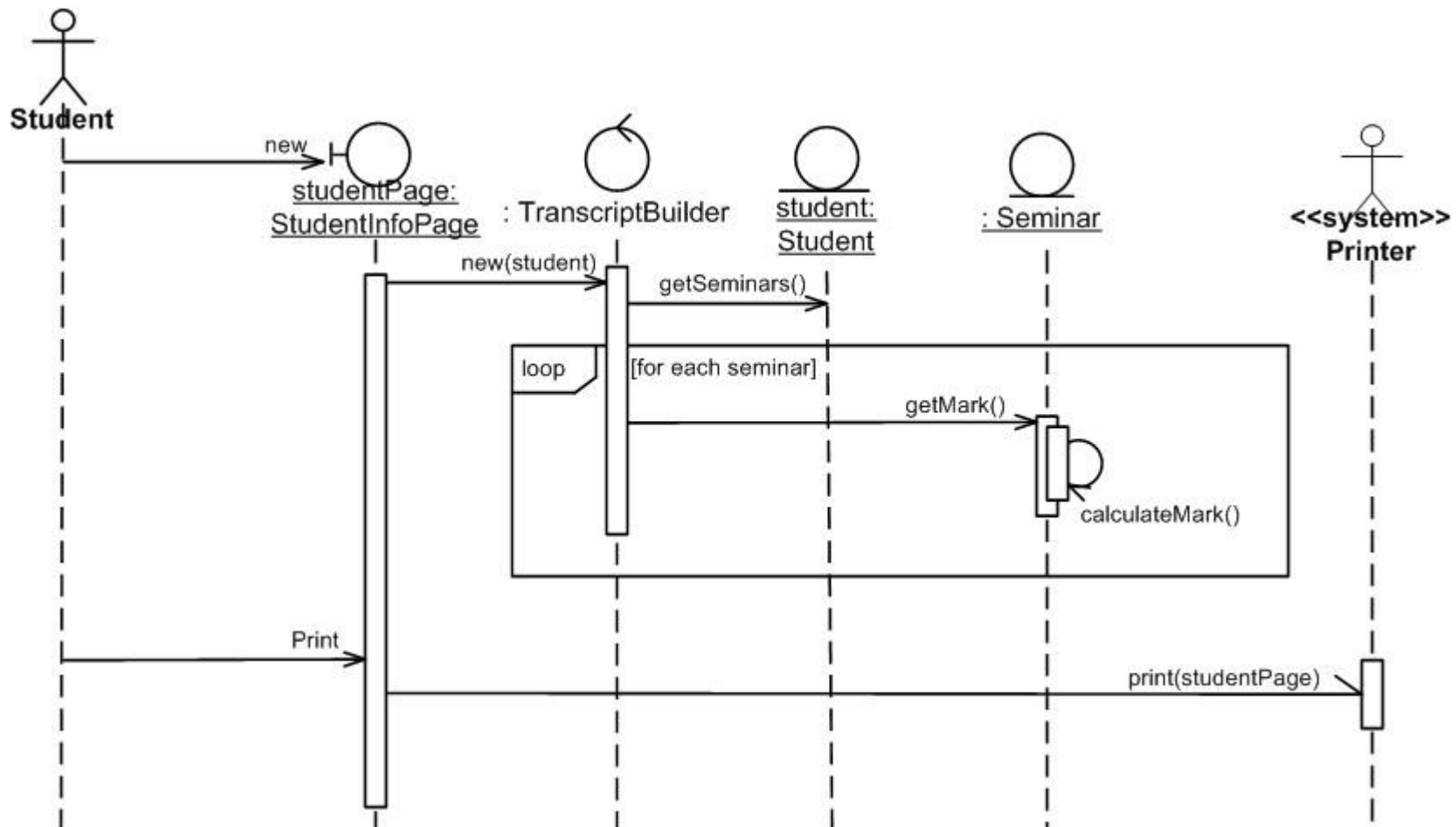
- למידע נוסף בנושא ניתן לפנות לאתר הבא:

<https://creately.com/blog/diagrams/sequence-diagram-tutorial/>

דוגמאות:



דוגמא:



מטלת כיתה

בנוסף לתיאור שירותי הספרייה (מהתרגול הקודם) יש להשתמש גם במידע הבא:

קוראים נרשמים לספרייה באמצעות ספרנית/ית, ואז הם נקראים "מנויים". עבור כל מנוי חדש יפתח כרטיס קורא עם כל המידע הרלוונטי כולל מידע אישי, מספר מנוי, שם משתמש, מספר טלפון נייד, ואימייל. הכרטיס כולל את היסטורית ההשאלות של המנוי, ובפרט, תקלות שונות בשימוש כגון איחורים בהחזרה או איבוד ספר.

מנהלת הספרייה יכולה להקפיא כרטיס קורא של מנוי ולמנוע ממנו שימוש בספרייה לתקופה זמנית (למשל, במקרה שהוא מחזיק ספר ולא מחזיר אותו).

המידע עבור כל ספר כולל: כותר (שם הספר), שם המחבר(ים), מס' מהדורה ותאריך ההדפסה, תחום (נושא, ספר מסוים יכול להשתייך למספר נושאים), תיאור מקוצר, מספר קטלוגי, מספר עותקים, תאריך רכישה, מיקומו על המדף, וטבלת תוכן העניינים סרוקה בפורמט של קובץ PDF.

Tasks to do:

1. Draw **activity** diagram for the “**Borrowing a book**” functionality from the previous tutorial (show using of swimlanes).
2. Construct complete **class** diagram of Library Information.
3. Present **sequence** diagrams for “Searching a book” functionality from the previous tutorial (show correspondence between class diagram and sequence diagrams).

Project Planning and Tools: Eclipse (JAVA projects, JAVADOC), GIT

