

מודלים לפיתוח מערכות תוכנה – תשע"ז סמ' ב Software Systems Modeling														
<table border="1"> <tr> <td>הרצאה</td><td>תרגיל</td><td>מעבדה</td><td>נקודות זכות (ני"ז)</td><td></td></tr> <tr> <td>3</td><td>0</td><td>0</td><td>3.0</td><td></td></tr> </table>					הרצאה	תרגיל	מעבדה	נקודות זכות (ני"ז)		3	0	0	3.0	
הרצאה	תרגיל	מעבדה	נקודות זכות (ני"ז)											
3	0	0	3.0											
12003				<b>קוד הקורס</b>										
○ הנדסת תוכנה (10014)				<b>דרישות קדם</b>										
אין				<b>דרישות מקבילות</b>										

#### מטרות הקורס

בקורס נלמד:

- (1) מבוא למידול
  - (2) טכניקות מידול (modeling) בסיסיות ופורמליות (מעט)
  - (3) הכרות עם כלים רלוונטיים
  - (4) פיתוח יכולת אבחנה ותיעדוף בין רמות שונות של הפשטה
  - (5) פיתוח יכולת פתרון בעיות וניתוח אנליטי ע"י מידול סוגים שונים של מערכות תוכנה
  - (6) פיתוח החשיבה לגבי מידול יעיל ואפקטיבי לצרכי ארכיטקטורה, תיכון (design), מימוש, אימות ותיקוף של מערכות תוכנה
- הקורס ילווה בתרגילים ופרויקט בהם ייבחנו ויישמו העקרונות השונים.

#### נושאי הקורס כוללים

UML

אימות פורמלי (OCL – לפי הזמן היכרות גם עם אחרים כגון Alloy, Z, MDE, Simulink)  
דוגמאות (case studies) למידול תוכנה  
תיכון וארכיטקטורת תוכנה, נושאים מתקדמים  
הצגת מחקרים ומודלים בנושאים קרובים של תהליכי פיתוח, פיתוח מונחה מודלים, בדיקות, הנדסה הפוכה (reverse engineering) ועקיבות.  
מידול וקוד פתוח.  
ייתכנו שינויים והתאמות במהלך הסמסטר.

#### חובות וציונים

תרגילים - 45%  
פרויקט סופי - 50%  
השתתפות פעילה בהרצאות ותרומה לקורס - 5% (אין חובת נוכחות למעט בהצגות נדרשות).

תרגילים בזוגות. פרויקט בקבוצות של עד ארבעה, כולל הצגה בכיתה.

את כל המודלים יש ליצור באמצעות כלי מידול רלוונטיים (ולא בכתב יד) ולהגיש תוצרים בעזרת מערכת בקרת גרסאות.  
מומלץ להביא מחשב נייד להרצאה.  
חובה להיות רשום למערכת הקורס והפורום באמצעות חשבון גיטהאב ולהיות מעודכן בתוכנו באמצעות מייל זמין וכדו'  
חלק מתוצרי הקורס יפותחו ברשיון של קוד פתוח

תרגילים:

- בקרת תצורה
- מידול עם UML
- תיאור ארכיטקטורת מערכת דרך זוויות שונות

פרויקט: בחירת פרויקט בקוד פתוח להערכה ותיאור, הערכת הפרויקט בהיבטים של מידול פונקציונלי ומימושי, הצגת הפרויקט בכיתה. העבודה תיכתב כפרק בספר כיתתי.

#### מרצה

ד"ר ראובן יגל, [robi@post.jce.ac.il](mailto:robi@post.jce.ac.il), שעות קבלה – בתיאום, פורום, שם משתמש ב-gh: robi-y

#### ספרות

מקור (5) ראו תקציר, יימסרו חומרי קריאה וקישורים למדריכים במהלך הסמסטר.

מאגר חומרים: <https://github.com/jce-il/sw-modeling-2017b>

#### ספרות מומלצת

- (1) A. Brown, G. Wilson, *The Architecture Of Open Source Applications*, Volumes I, II, lulu.com (2012, 2014)
- (2) M. Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, 3e, Addison-Wesley (2003).
- (3) M. Brambilla, J. Cabot, M. Wimmer, *Model-Driven Software Engineering in Practice*, Morgan & Claypool (2012).
- (4) R.S. Pressman, *Software Engineering: a Practitioner's Approach*, 7<sup>th</sup> ed., McGraw-Hill (2010).
- (5) Rozanski and Woods, *Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives*, 2<sup>nd</sup> ed, Addison-Wesley Professional (2011)  
Reference summary: [http://www.viewpoints-and-perspectives.info/vpandp/wp-content/themes/secondedition/doc/registered/VPandP\\_Reference.pdf](http://www.viewpoints-and-perspectives.info/vpandp/wp-content/themes/secondedition/doc/registered/VPandP_Reference.pdf)
- (6) I. Sommerville, *Software Engineering*, 9<sup>th</sup> ed., Addison Wesley (2011).
- (7) E. M. Clarke, Jr., O. Grumberg & D. Peled, *Model Checking*, MIT Press (1999).
- (8) D. Peled, P. Pelliccione, P. Spoletini, "Model Checking", In: *Wiley Encyclopedia of Computer Science and Engineering*, Wiley (2009).
- (9) B. Berard ... [et al.], *Systems and Software Verification: Model-Checking Techniques and Tools*, Springer (2001).
- (10) E. Evans. *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Prentice Hall (2003).
- (11) Wilson, *Architecture of Open Source Projects*