

מודלים לפיתוח מערכות תוכנה – תשע"ו סמ' ב Software Systems Modeling				
הרצאה	תרגיל	מעבדה	נקודות זכות (נ"ז)	
3	0	0	3.0	
קוד הקורס				12003
דרישות קדם				○ הנדסת תוכנה (10014)
דרישות מקבילות				אין

מטרות הקורס

בקורס נלמד:

- (1) מבוא למידול
 - (2) טכניקות מידול (modeling) בסיסיות ופורמליות (מעט)
 - (3) הכרות עם כלים
 - (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 תיכון (design), מימוש, אימות ותיקוף של מערכות תוכנה
- הקורס ילווה בתרגילים ופרויקט בהם ייבחנו ויישמו העקרונות השונים.

נושאי הקורס כוללים

UML (כולל OCL – לפי הזמן היכרות גם עם MDE, Z, Alloy, 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.
ייתכנו (=צפויים) שינויים והתאמות במהלך הסמסטר.

חובות וציונים

חובת רישום למערכת הקורס (שם משתמש בגיטהאב וג'יטר)
אין חובת נוכחות למעט בהצגות נדרשות, אך יינתן בונוס 5% על השתתפות פעילה.
תרגילים - 50%
פרויקט סופי - 50%
אפשר להגיש בקבוצות של 2-3 סטודנטים, אך בכל משימה יש להחליף צוות! כל קבוצה נדרשת לפחות פעם אחת במהלך הסמסטר להציג את עבודתה על הפרויקט בסקר כיתתי.
את כל המודלים יש ליצור באמצעות כלי מידול רלוונטיים (ולא בכתב יד) ולהגיש תוצרים בעזרת מערכת בקרת גרסאות.

תרגילים:

- בקרת תצורה
- מידול עם UML
- פיוח מונחה מודלים \ DDD
- עוד (TBD)

פרויקט (תכנית כללית):

- שלב 1 – בחירת פרויקט בקוד פתוח להערכה ותיאורו
- שלב 2 – הערכת הפרויקט בהיבטים של מידול
- שלב 3 – הצגת הפרויקט בכיתה
- שלב 4 – סיכום כפרק בספר כיתתי

מרצה

ד"ר ראובן יגל, robi@post.ice.ac.il, שעות קבלה – בתיאום, פורום, שם משתמש ב-gh: robi-y
חומרי הקורס:

ספרות

אין ספר קורס נדרש, יימסרו חומרי קריאה וקישורים למדריכים במהלך הסמסטר.

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*, 7th ed., McGraw-Hill (2010).
- (5) I. Sommerville, *Software Engineering*, 9th ed., Addison Wesley (2011).
- (6) E. M. Clarke, Jr., O. Grumberg & D. Peled, *Model Checking*, MIT Press (1999).
- (7) D. Peled, P. Pelliccione, P. Spoletini, "Model Checking", In: Wiley Encyclopedia of Computer Science and Engineering, Wiley (2009).
- (8) B. Berard ... [et al.], *Systems and Software Verification: Model-Checking Techniques and Tools*, Springer (2001).
- (9) E. Evans. *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Prentice Hall (2003).