תוכן הגליון

מאמר המערכת 2

לזכרו של ד"ר עוזי ערמון (קסט) 3

שימוש קונסטרוקטיביסטי בטעויות בהוראת יסודות מדעי המחשב

רונית שמלו ודוד גינת 6

על הורשה מרובה והורשה בכלל

אודי מלכה 17

סקר אתר האינטרנט של המרכז הארצי 24

פרויקט רמזור למורה 27

סמינר קיץ למורים מובילים תשע"ד 28

טופס משוב 33

קראתם את העיתון?
אל תשכחו למלא את טופס המשוב
הנמצא בעמוד האחרון
ולשלוח אותו אל מינהלת מל"מ.

כתובתנו:
מרכז המורים הארצי למדעי המחשב
המחלקה לחינוך למדע וטכנולוגיה, טכניון, חיפה 32000
טלפון 04-8292880 , פקס 04-8293004

מאמר מערכת

קוראים יקרים

לפניכם הגליון הראשון של "הבטים בהוראת מדעי המחשב" לשנת תשע"ה. אתם מוזמנים לקרוא אותו באתר ללא תשלום ולאחר מכן למלא את טופס המשוב ולשלוח אותו למינהלת מל"מ. שליחת דפי המשוב היא התנאי להמשך קיומו של העיתון.

מה בגיליון הפעם?

ד"ר עוזי ערמון (קסט) נפטר מדום לב פתאומי ב- 20 ביולי 2014. עוזי עסק בהוראת מתמטיקה ומדעי המחשב עשרות שנים, לימד במכללת אוהלו בקצרין, במכללה להוראה בסכנין, במכללת עמק הירדן, בטכניון, באוניברסיטת חיפה ובמכללת תל חי. רוב עבודתו הוקדשה להכשרת מורים ומורים רבים למדעי המחשב הם תלמידיו לשעבר. בפתח הגליון נביא דברים לזכרו שנכתבו על-ידי משפחתו, חבריו לעבודה ותלמידיו לשעבר. יהי זכרו ברוך.

המאמר של רונית שמלו ודוד גינת מוקדש ל**שימוש קונסטרוקטיביסטי בטעויות בהוראת יסודות מדעי המחשב**. המאמר מציג גישה שהם פיתחו, של למידה מטעויות ביסודות מדעי המחשב, בהוראת OOP. הגישה מבוססת על אוסף של עקרונות, הכוללים: קונפליקט קוגניטיבי, מאפיינים קונסטרוקטיביסטיים, ייחוס שגיאה, הסבר עצמי, רפלקציה ועידון דימוי מושג של מונחי OOP בסיסיים. במאמר מוצגת הגישה, סיכום של תוצאות המחקר יחד עם דעותיהם של הסטודנטים בנוגע לגישה המיושמת.

המאמר של אודי מלכה, **על הורשה מרובה והורשה בכלל**, עוסק בדרך שעבר עם התלמידים ממטלה בה התבקשו התלמידים ליישם בשפת התכנות C++ תרגיל בנושא הורשה מרובה ועד הרהורים ותובנות שעלו בכיתה מדיונים וחקר אודות מנגנון ההורשה בכלל והאלטרנטיבות שלו בשפות Java ו- C#.

בחודשים ספטמבר 2014 – דצמבר 2014 קיימנו סקר של אתר האינטרנט של מרכז המורים הארצי למדעי המחשב. את תוצאות הסקר וניתוח שלהן תוכלו למצוא בהמשך.

תוכלו למצוא בגיליון גם קול קורא לפיתוח מערכי שיעור בפרויקט "רמזור למורה" של "קשר חם", וכן דיווח קצר על סמינר הקיץ למורים מובילים שהתקיים בניר עציון ביולי 2014.

מומלץ לעקוב אחר ההודעות השוטפות על פעילויות המרכז הארצי באתר האינטרנט שלנו http://cse.proj.ac.il.
אתם מוזמנים להמשיך לגלוש באתר, לקרוא ולהוריד קבצים.

נשמח גם לקבל מכם חומרים מפרי עטכם בכל נושא שיכול לעניין את קוראי העיתון.

קריאה מהנה,

ממערכת העיתון ומצוות המרכז הארצי

ד"ר עוזי ערמון (קסט) ז"ל

בצהרי יום ראשון, כ"ב בתמוז תשע"ד, 20 ליולי 2014, נדם לבו של עוזי ערמון (קסט) באופן פתאומי בעת שהתכונן לשנה האקדמית, בחדר העבודה שלו אל מול נוף הכנרת במושב אמירים.

עוזי נולד באוגוסט 1947 בתל אביב, שם עברו עליו ילדותו ונעוריו. לאחר שחרורו מהצבא, החל ללמוד מתמטיקה ופיזיקה באוניברסיטה העברית בירושלים שם סיים תואר ראשון, שני (נושא התיזה: בעיות גדירות בלוגיקה מתמטית) ותעודת הוראה. עם סיום הלימודים בירושלים עבר עוזי עם משפחתו למושב אמירים.

תמיד האמין שייעודו להיות מורה. אכן עוזי מימש ייעוד זה והחל ללמד מתמטיקה בתיכון העיוני בצפת, סחף אחריו תלמידים רבים והתייחס אליהם כאל ילדיו. עם אין סוף סובלנות, מבט אוהב ומאמין ביכולתם להצליח, הוא היה לרבים מתלמידיו הרבה מעבר למורה למתמטיקה ומהר מאד הממוצע הבית ספרי בבגרות במתמטיקה עלה משמעותית.

בשנת השבתון המשיך ללימודי דוקטורט במחלקה להוראת המדעים בטכניון, בהנחיית פרופ' אורי לירון (נושא התיזה: אלגוריתמים אינטרינזיים לייצוג עקומים מתמטיים ב- Logo), ומשם לפוסט דוקטורט באוניברסיטת אלינוי בשיקגו.

בהמשך השתלב כמרצה באקדמיה לימד קורסים רבים בתחומים שונים ומגוונים. בשביל הגיוון, האתגר והעניין הוא הקפיד כל שנה, ללמד קורס חדש. עם השנים והפרויקטים הרבים שהיה מעורב בהם, עוזי צבר ניסיון רב בתחומי החינוך לחשיבה, היצירתיות ובהקניית אסטרטגיות למידה.

בין היתר, עוזי היה מהמייסדים של החוגים להוראת המתמטיקה ומדעי המחשב במכללת אוהלו בקצרין ולימד במכללה להוראה בסכנין, במכללת עמק הירדן (כנרת), באוניברסיטת חיפה (החוג למתמטיקה ומדעי המחשב), ובמכללת תל חי.

בשנים האחרונות עוזי עסק בעיקר בהכשרת מורים ולימד את המורים לעתיד איך ללמוד ואיך ללמד ללמוד. עוזי השקיע בתלמידיו את מיטב מרצו, עם המון נתינה, הקשבה, אכפתיות ואמונה שלמה ביכולות של תלמידיו ולא רק בתחום הנלמד.

בנוסף להוראה, עסק עוזי בפיתוח תכניות לימודים, הקמת חוגים חדשים, השתתף בארגון כנסים, וכתב מספר ספרי לימוד.

עוזי השאיר אחריו אישה, שלומית, 4 ילדים ו- 8 נכדים שכל אחד ואחת מהם קיבל תשומת לב ואהבה אין-סופית.

יהי זכרו ברוך.

דברים לזכרו של ד"ר עוזי ערמון

עוזי היה דמות משפיעה ומרכזית במערכת החינוך למתימטיקה ומדעי המחשב, והשאיר חותם עמוק אצל מאות תלמידים ומורים. הידיעה על מותו הפתאומי בטרם עת היכתה את כולנו בתדהמה וצער.

זכרונותי על עוזי שייכים בעיקר לתקופה בה למד ועבד בטכניון, ובה גם כתב את עבודת הדוקטורט בהנחייתי. לעוזי היתה סקרנות אינסופית וידע רב ומעמיק בנושאי חינוך, מדעי המחשב ומתימטיקה, והוא הצטיין בשילוב נלהב ומעמיק של כל שלושת התחומים. והעבודה אתו היתה תמיד מרתקת ומאתגרת. אני זוכר דיונים מעמיקים על הוראת רקורסיה, ועל המתימטיקה שמאחורי "גיאומטריית הצב"; למשל, למה קל מאוד "ללמד את הצב" (הז'רגון המקובל בימים ההם לתיכנות בשפת לוגו) איך לשרטט מעגל, אבל קשה מאוד ללמדו איך לשרטט אליפסה. (יש לזכור שהאסתטיקה של תיכנות בשפת הצב העדיפה פעולות צב "אוטנטיות" – לך קדימה או אחורה, פנה ימינה או שמאלה – אבל לא שימוש במערכת צירים קרטזית.) בהמשך לספרים של פאפרט ושל אבלסון ודיססה, עוזי העמיק בעבודת הדוקטורט בחקר ה"מתימטיקה של הצב", ואף פרסמנו במשותף שני מאמרים בנושא זה בז'ורנלים מקצועיים בחו"ל. (*)

התלהבותו ומסירותו למקצוע החינוך, הידע המעמיק שלו, וחיוכו המאיר יחסרו לנו מאוד.

פרופ' אורי לירון

המחלקה לחינוך למדע וטכנולוגיה, טכניון

את עוזי הכרתי כשהחל ללמד במכללת אוהלו לפני מספר שנים. איש חייכן וצנוע וכשדיבר על מתמטיקה נצצו עיניו בלהט ושמחה מתוך התלהבות גדולה. עוזי נתן דוגמה אישית לדרכי למידה ישירה ועקיפה. בכל שבוע תלה עוזי על לוח מודעות במכללה חידה שבועית שבמהלך השבוע לפתור. חידות ברמה התואמת לימודים לתואר ראשון. דיבורו היה שקול ומדוד והשרה אוירה נעימה מאוד לכל סובביו. דמות מופת לחיקוי. יהי זכרו ברוך.

ד"ר איל ויסבליט

מכללת אוהלו

את עוזי הכרתי כאשר הגעתי למחלקה לחינוך למדע וטכנולוגיה בטכניון (או בשמה הקודם, המחלקה להוראת הטכנולוגיה והמדעים). למדתי לתעודת הוראה (במקביל ללימודי התואר הראשון במדעי המחשב) ועוזי היה דוקטורנט במחלקה וחלק מצוות לוגו של ה"סטודנטים של אורי". כאשר בחרתי באורי לירון כמנחה לתואר השני, התקבלתי לצוות הזה בחיבוק חם ובמיוחד קיבל אותי עוזי בזרועות פתוחות. זכיתי לעבוד לצידו של עוזי במסגרת אל"י – אגודת לוגו הישראלית, בהוצאה לאור של "צו לצב" (עלון אל"י), ובתרגול בקורסים במחלקה.

למדתי מעוזי להבין טוב יותר את גרפיקת הצב, למדתי ממנו על ציקלואידות ויצורים גיאומטריים שונים ומשונים אחרים, למדתי ממנו על ייצוג אינטרינזי (פנימי) ואקסטרינזי (חיצוני), ובעיקר למדתי ממנו שלא תמיד צריך לענות מיד לתלמיד ששואל שאלה ושיש יתרון בהשהיה הזו כי יש סיכוי טוב שהתלמיד עצמו יצליח לענות לשאלה ששאל. זכורים לי אפילו מספר ויכוחים שהיו לנו על השאלה "עד מתי המורה צריך לשתוק?". לא בכל הפעמים הסכמתי לגישות הרדיקליות של עוזי בנושא הזה, אבל תמיד ידעתי שאפשר ללמוד ממנו ושתהיה לו סבלנות אינסופית ויכולת הקשבה והכלה מופלאה.

זכור לי במיוחד הכנס הבינלאומי הראשון שהשתתפתי בו (כנס LME במונטריאול, קנדה). עוזי ליווה אותי במהלך כל הכנס ודאג שלא אלך לאיבוד, הכיר לי את חברי הקהילה הבינלאומית, וסייע לי לקליטה מהירה וקלה.

קשה לדבר על עוזי בלשון עבר. אני אזכור אותו כאדם נעים הליכות, צנוע, אוהב ומכבד את הזולת, איש חינוך ייחודי ומקצוען של ממש. יהי זכרו ברוך.

ד"ר תמי לפידות

מנהלת מרכז המורים הארצי למדעי המחשב

(*) "How to explain a cycloid to a turtle?" *The Journal of Computers in Mathematics and Science Teaching*, 7(3) (1988), 55-58.
"Turtle goes to college: intrinsic representations and graphical integration". *International Journal of Mathematics Education in Science and Technology* 22,4 (1991), 577-584.

עוזי ערמון (קסט) ז"ל 1947 - 2014

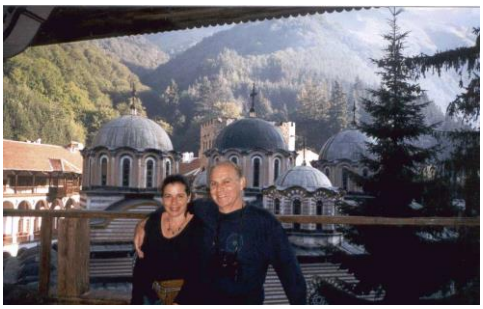
בתאריך כ״ב בתמוז תשע״ד, 20 ביולי 2014, נפטר בטרם עת המנחה שלי דר׳ עוזי ערמון. בשנותי הראשונות כמורה החלטתי להשלים את התואר הראשון ולמדתי, בשנת 1990, את הקורס שיטות הוראה. בקורס נחשפתי לגישתו המיוחדת של עוזי לאנשים ובמיוחד ללומדים. שם התחלתי לחקור את דרכו המיוחדת של עוזי כיצד ללמוד וכיצד לסייע ללמידה.

עוזי שהיה מראשוני מורי הלוגו בישראל משנת 1980 –ועד מותו הפתאומי הצליח ליצור בשנת 1991 את התוכנה לה קראנו בשם: ״מולטי-טכנו-לוגו״. עוזי ייצר בתוך סביבת תמלילוגו (LogoWriter 2) את הכלים של תוכנת ה-LEGO-Logo. טבעה הפתוח של תמלילוגו איפשר את יצירת כלי הבקרה על עולם חיצוני, רובוטים או מערכות הנדסיות מבוקרת מחשב שלומדים יצרו. העושר התכנותי של הסביבה החדשה היה גדול מסכום חלקיה.

ניתן היה בקלות רבה ללמד ילדים בכל גיל ובכל רמה לימודית לתכנת בקרה אוטונומית של מכונית שנעה בין שני מחסומים פיסיים ובמקביל מכונית שנעה על המסך בין שני מחסומים אופטיים. בעת הגעתה של המכונית הוירטואלית למחסום האופטי היא הייתה משנה כיוון אך גם גורמת למכונית הפיסית לשנות כיוון ולהפך, אם המכונית הפיסית הגיעה למכשול הפיסי ראשונה היא שינתה את כיוונה אך גם גרמה למכונית הוירטואלית לשנות את כיוון תנועתה.

גולת הכותרת של עבודתנו המשותפת הייתה בשנת 1993, כאשר לקחתי את התוכנה שעוזי פיתח ״מולטי-טכנו-לוגו״ ויישמתי אותה לפי עקרונות ההוראה היצירתית של עוזי בסביבת רובוטיקה בחטיבה העליונה. בקורס שקראנו לו: ״מלמי״ד: **מ**נוף **ל**קידום **מ**וטיבציה **וי**דע, נולדה מגמת המכטרוניקה הראשונה בישראל שהתפשטה בשנות ה- 2000 ועד היום למאות בתי ספר.

השילוב של מומחיות להוראת מתמטיקה ומומחיות להוראת מדעי המחשב הביאו את עוזי ליצירת שיטות מקוריות להוראת נושאים רבים מתחומי ידע אלו. עוזי לעולם לא לימד קורס אחד שנים רבות, הוא אהב מאוד ללמוד בעצמו דברים חדשים ולנסות ללמד אותם אחרים. חלק משיטותיו תועדו בחומרי לימוד רבים אותם כתב ובספרים שהוציא לאור. ואולי המפורסם שבהם הוא מבוא למדעי המחשב עם c++ בו למעשה ייצר את גרפיקת הצב של שפת לוגו בתוך שפת C כאמצעי להוראת מתמטיקה ורקורסיה.



עוזי אהב את כל ישראל ואת ירושלים, בה למד את שני תאריו הראשונים במיוחד. אולם כאשר גילה בזכות שלומית, אשתו, את אמירים התאהב בגליל כולו ובאמירים בפרט. ביתו, הנשקף אל הכינרת, הפך למרכז שואב לצעירים, בכל גיל, שבאו לשמוע את עוזי. בביתו שבאמירים גם הקים את צוות לגו-לוגו בגליל שפעל במכללת תל חי, שם לימד עוזי קורס לגו-לוגו להורים וילדים, לתלמידים בעלי מוגבלויות ותלמידים מחוננים.

החיים האישיים שלי השתלבו במשפחתו של עוזי הן כאשר הצטרפנו לטיולים ברחבי הארץ והן כאשר נפלה בידי הזכות להיות מורה של אחד מבניו. בתקופות רבות של חיי הייתי נוסע לאמירים רק כדי שאוכל לשבת בבית הנפלא של משפחת ערמון ולשמוע את עוזי מדבר על כל דבר שקשור לעולם החינוך, המשפחה או לעולם הטיולים.

גם ברגעים אלו בהם אני כותב על עוזי בלשון עבר אני לא מעכל שלעולם לא אוכל להתייעץ איתו יותר. אני משוכנע שהליכתו של עוזי מעולמנו הותירה חלל עצום בקרב כל מי שהכירו וחלל עמוק בלתי נתפס במשפחתו הקרובה והנרחבת.

זכרו צרוב לנצח בקרב תלמידיו הרבים!

ד״ר ירון דופלט

מפמ״ר הנדסת מכונות, משרד החינוך

שימוש קונסטרוקטיביסטי בטעויות בהוראת יסודות מדעי המחשב

רונית שמלו

המכללה להנדסה SCE
וקבוצת מדעי המחשב
המחלקה להוראת המדעים
אוניברסיטת תל-אביב

דוד גינת

קבוצת מדעי המחשב
המחלקה להוראת המדעים
אוניברסיטת תל-אביב

המאמר(*) מציג גישה שפיתחנו, של למידה מטעויות ביסודות מדעי המחשב, בהוראת OOP. הגישה מבוססת על אוסף של עקרונות, הכוללים: קונפליקט קוגניטיבי, מאפיינים קונסטרוקטיביסטיים, ייחוס שגיאה, הסבר עצמי, רפלקציה ועידון דימוי מושג של מונחי OOP בסיסיים. הגישה מוצגת וחלק מהפעילויות בה מוצגות גם הן בצורה מובנית. הפעילויות שפותחו מכוונות לטיפול בתפיסות שגויות שנצפו במחקר קודם שלנו, שפורסם בין היתר בגיליון ינואר 2013 בהיבטים [16]. יישמנו את הגישה על סטודנטים במכללה להנדסה וערכנו מחקר השוואתי עם קבוצת ניסוי וביקורת. התוצאות היו חיוביות. אנו מציגים סיכום של תוצאות המחקר יחד עם דעותיהם של הסטודנטים בנוגע לגישה המיושמת.

1. מבוא

טעויות קשורות לעשייה שגויה. כל אחד טועה, ואפשר ללמוד מטעויות, לא פחות, ואולי אף יותר, מאשר מנכון.

במאמר זה, אנו מציגים מחקר של למידה קונסטרוקטיביסטית מטעויות. מטרת המחקר היא לבחון דרך הוראה חדשה המשלבת התייחסות מפורשת לטעויות ולימוד מהן בהוראת אבני הבניין של הפרדיגמה המונחית עצמים במדעי המחשב.

במדעי המחשב מקובל ללמוד מטעויות, בהקשר של ניפוי שגיאות (debugging) בפיתוח תוכנית. תהליך ניפוי השגיאות נחשב כתהליך של "ניקוי" תוכנית מבאגים (bugs), על מנת שהתוכנית תרוץ בהצלחה.

ספרי לימוד במדעי המחשב מציינים בדרך כלל את השלב החשוב של איתור באגים, ללא התמקדות באופן מפורש בלמידה מטעויות. עם זאת, שגיאות הן לא בהכרח באגים בתכנות. הן יכולות לנבוע מהבנה רעיונית מוגבלת, חוסר הבנה של מושגי יסוד או פתרון בעיות באופן שגוי. פרט לניסיונות של Ginat [6,7] לא נעשו מחקרים הכוללים שימוש מפורש בשגיאות כמניע ללמידה במדעי המחשב. במאמר זה, אנו מרחיבים את נקודת המבט של לימוד מטעויות, באמצעות שימוש בטעויות לפיתוח ידע קוהרנטי ביסודות מדעי המחשב.

(*) מבוסס על מאמר שהתקבל לכנס SIGCSE 2013, בהתבסס על עבודת דוקטורט. את הגרסה האנגלית של המאמר ניתן לקרוא ב-

Ginat, D. & Shmallo, R. (2013). Constructive Use of Errors in Teaching CS1. In Proceedings of the 44th ACM technical symposium on Computer Science Education. pp. 353-358. ACM.

בשני העשורים האחרונים, נערכו מחקרים שונים במתמטיקה, בפיזיקה ובתחומי ידע שונים של שימוש בטעויות כאמצעי ללמידה. Ohlsson [12] מציע תיאוריה, שעוסקת בזיהוי ותיקון שגיאות על ידי אנשים, באמצעות תרגול מיומנויות. שגיאה גורמת לקונפליקט בין הידע אשר הלומד מאמין שהוא אמת לבין מה שהלומד תופס כמצב הנוכחי. הוא העלה השערה שטעויות נגרמות בעיקר כתוצאה מהכללה (overgeneralization) וממורכבות של מבנה ידע כלליים. לטענתו, על המשימות שהלומד מקבל, ליצור קונפליקט קוגניטיבי שיעורר אצל הלומד תהליך של מחשבה (blame assignment), שיוביל את התלמיד לשיוך השגיאה. כלומר, הבנת מקור הטעות. הדבר בהכרח יוביל לבדיקה מחדש של הידע (knowledge revision) אצל הלומד.

Borrasi [1] זיהתה את הפוטנציאל של ההתייחסות לשגיאות ופיתחה מסגרת של ניצול טעויות כקרש קפיצה לחקירה על טיבה של המתמטיקה. היא הציעה גישת הוראה המבוססת על חקר בעיה, טעויות, קונפליקט, ספק ורפלקציה, כדי לחזק ולשכלל את הגדרות המושגים המתמטיים ואת תהליך פתרון הבעיות. Melis et al. [10,11] בחנו את הערך הפדגוגי של דוגמאות שגויות. הם הציגו לתלמידים דוגמאות שגויות באופן ישיר ותוך שימוש במערכת ActiveMath, שהיא סביבת למידה מבוססת אינטרנט עבור הוראת מתמטיקה. היא ועמיתיה הבחינו בתרומה הקוגניטיבית (מתמטית) והמטה-קוגניטיבית (ניתוח) של פעילות מסוג זה.

בפיזיקה, Yerushalmi & Polingher [19] חילקו לתלמידים דפי עבודה המכילים היגדים מוטעים, "דף מלא טעויות", שנאספו במהלך השנים מתשובות שגויות נפוצות של תלמידים וביקשו מהם לאתר בכל היגד את הטעות ולשכנע את כותב ההיגד שהוא טעה. מטרתם הייתה לחזק אצל התלמידים את התפיסה של תהליך הפתרון כמעשה הכרוך בבדיקה עצמית ואת הנטייה לבצע בדיקה כזו. Pinkerton [13] הנהיג "סיבוב שני" לבחינה תוך שהוא מבקש מתלמידיו לזהות, להסביר ולתקן את השגיאות הקונספטואליות בפתרונות הבחינה המקוריים שלהם.

Siegler [17] ו-Grobe & Renkl [8] בדקו האם הסברים עצמיים עבור פתרונות נכונים ושגויים יעילים יותר מאשר עבור פתרונות נכונים בלבד. הם מצאו שכאשר תלמידים למדו והסבירו באופן עצמי גם פתרונות נכונים וגם שגויים, הדבר הוביל לרמת דיון גבוהה יותר וללמידה טובה יותר. לטענתם אין זה יעיל לשלב דוגמאות עם שגיאות ללומדים מהתחלה. רצוי תחילה להראות ללומדים פתרונות נכונים ולאחר מכן להראות ולעסוק בפתרונות השגויים, המסייעים להעמקת הידע שכבר נרכש קודם לכן.

במחקרים שתוארו לעיל, של למידה מטעויות, כרוך היבט של קונסטרוקטיביזם, כי הלמידה מבוססת על ידע קודם של התלמידים, שהוא לא מדויק, או מעורפל במידה מסוימת. המטרה היא לשפר את הידע והכישורים של התלמידים על ידי יצירת קונפליקט קוגניטיבי באמצעות טעויות. מחקרים קונסטרוקטיביסטיים תומכים ברעיון של קונפליקט כזרז ללמידה. קונפליקט גורם לבחינת הידע והנהלים אצל הלומד [2].

במאמר זה, אנו מציגים גישה קונסטרוקטיביסטית ללימוד באמצעות טעויות, בפיתוח וניתוח של תוכניות, הכוללות שימוש במושגי יסוד של הפרדיגמה המונחית עצמים. הגישה שלנו מבוססת על מחקרים קודמים שנעשו אודות לימוד מטעויות, שתוארו לעיל, והיא בנויה על קשיים של תלמידים עם מושגי היסוד בפרדיגמה המונחית עצמים.

יישמנו את הגישה שלנו בכיתה בת 52 סטודנטים לתואר ראשון, בקורס יסודות מדעי המחשב בשפת Java. המחקר שערכנו היה מחקר השוואתי, שבו הביצועים של סטודנטים אלו הושוו עם קבוצה אחרת בת 47 סטודנטים שלמדו את אותם חומרי הלימוד, אבל בדרך המסורתית ולא בדרך של לימוד מטעויות. שתי הקבוצות, קבוצת הניסוי וקבוצת הביקורת היו דומות במונחים של יכולות ורקע הן במדעי-המחשב והן במתמטיקה. בפרק הבא, פרק 2, אנו מציגים ומדגימים את הגישה שלנו ללימוד מטעויות. בפרק 3, אנו מציגים את המתודולוגיה של המחקר ההשוואתי שלנו ואת תוצאות המחקר, כולל את נקודת הראות של התלמידים אודות הגישה של לימוד מטעויות.

בפרק 4, אנו דנים בתוצאות שהתקבלו ומשמעותן להוראה.

2. למידה מטעויות בתכנות מונחה עצמים

תכנות במדעי המחשב, כולל אלמנטים עיקריים שונים, ביניהם: הבנת התכונות של שפת תכנות, הבנה של מודל המחשב, ניתוח קלט/פלט, מידול של "העולם" ועיצוב אלגוריתמי. תלמידים מתחילים מגלים קשיים, מסוגים שונים, עם כל האלמנטים הללו. בגישה שלנו החלטנו להתמקד באלמנט הראשון ברשימה שהצגנו, שהוא – הבנת מרכיבים יסודיים של שפת התכנות. שפת התכנות היא Java, והיא שייכת לפרדיגמה המונחית עצמים. המיקוד שלנו קשור גם, במידה מסוימת, להבנה של מודל המחשב.

מחקרים קודמים של קשיי מתחילים חשפו הבנה מוגבלת של המונחים הבסיסיים ביותר של הפרדיגמה המונחית עצמים, כולל המושגים: מחלקה, עצם ובנאי (למשל, [4,5,9,14]). מחקר שלנו, שנערך לאחרונה [16], חשף תפיסות מעורפלות של מתחילים, אודות מושגים אלו ואחרים כדוגמת "משתנה סטטי". בנוסף, המחקר חשף נטיות של מתחילים להרחיב ו/או לצמצם תכונות, הקשורות למושגים אלו. לדוגמא, מתחילים תופסים משתנה סטטי כישות שיכולים להיות לה מופעים רבים בזיכרון, וכך הם מרחיבים את מספר המופעים שלה. אחרים אינם מודעים לצורך בפנייה לבנאי בעת יצירת אובייקט ובכך הם מקטינים או מפשטים מדי את המנגנון של יצירת האובייקט. חלק מהמתחילים מודעים לתפקיד החיוני של הבנאים, אבל מאמינים שרק בנאי אחד יכול להיות מוגדר במחלקה ובכך הם מקטינים את מספר הדרכים האפשריות להגדרת בנאי.

הגישה שלנו ללמידה מטעויות של מושגי OOP בסיסיים

ממצאים במחקר קודם שלנו [16] מרמזים על מספר הנחות:

- עבור מספר לא קטן של מושגי יסוד בסיסיים, דימוי המושג [18] אינו זהה להגדרה המדויקת של המושג. חוסר הדיוק בהגדרה הוא ההיסט של ההרחבה או ההצרה של תכונות המושג בפועל.
- חלק מהתפיסות המוטעות נובעות מדוגמאות ספציפיות, המוצגות למתחילים, שכתוצאה מהן הם משתמשים באנלוגיות לא נכונות ובהכללות שגויות.
- חלק מתכונות המושג נתפסות כאוסף של פעולות יותר מאשר תכונות המוגדרות היטב. הדבר מייצג הסתכלות אופרטיבית יותר מאשר דקלרטיבית על התכונות.

בעקבות ההנחות הללו והמחקרים אודות לימוד מטעויות שהצגנו קודם לכן, פיתחנו **מודל ללמידה מטעויות של מושגי OOP בסיסיים.** המודל כולל את המרכיבים הבאים:

- **זיהוי תפיסת התלמיד השגויה.** יש לזהות את התפיסה השגויה של התלמיד אודות מושג היסוד בפרדיגמה.
- **הגדרת הפער בהתייחס למושגים של הרחבה ו/או הצרה.** בהתייחס לתפיסה השגויה שזוהתה, יש לקבוע האם חוסר הדיוק בהגדרה נובע מהרחבה או מהצרה של תכונות המושג בפועל. הפער בין הנכון לבין תפיסת התלמיד מגדיר את ההצרה או ההרחבה של המושג.
- **הגדרת המשימות לצמצום הפער ועידון התפיסה (Tasks that refine conceptions).** מטרת המשימות בגישה הקונסטרוקטיביסטית היא לבצע שיוף של התפיסה השגויה (במחקר שלנו הוספת החלק שהוצר או הורדת החלק שהורחב).
- **הפורמט של המשימות (Task format).** אופי המשימות הוא אחד משלוש הצורות הבאות: 1. תיאור של תוצאת הביצוע של קוד שניתן. 2. הערכה של היגד אודות תכונות מושג או על הצהרה הניתנת לתכנות. 3. משימת פיתוח קוד ("מהתחלה" או מקוד חלקי שניתן).
- **פעילויות כיתה (Class activities).** תהליך משימת הפתרון, שבסופו של דבר מוביל לבדיקה מחדש של הידע, מתבצע באחת או יותר מהדרכים הבאות: 1. כל תלמיד פותר את המשימה לבד. 2. זוג סטודנטים פותר את המשימה. 3. מתנהל דיון בכיתה.

- **הסברים עצמיים (Self-explanation).** במהלך הפעילות של סטודנטים בודדים או זוגות סטודנטים יש לעודד הסברים עצמיים, מתוך מחשבה שכאשר לומד מסביר לעצמו את הפתרון הנכון הוא נוכח בטעות של עצמו ומבין מה הוא עשה לא נכון.
- **יצירת קונפליקט קוגניטיבי (Cognitive conflict).** תפקיד המשימות לשפר את הידע והכישורים של התלמידים על ידי יצירת קונפליקט קוגניטיבי, באמצעות טעויות.
- **שיוך השגיאה (Error attribution).** הנקודה שבה הקונפליקט הקוגניטיבי מתעורר צריכה לעורר תהליך של מחשבה, שמוביל את התלמיד להבנה של מקור הטעות ולבדיקה מחדש של הידע על ידי חידוד של דימוי המושג אצל הלומד.
- **רפלקציה (Reflection).** הדיון בכיתה מסתיים במבט לאחור על תהליך הפתרון. בדיון על התפיסות השגויות, יש לעודד את התלמידים להציע סיבות אפשריות לטעויות. כמו כן, הרפלקציה מסייעת בפיתוח תפיסות דקלרטיביות-הצהרתיות **(Declarative conceptions)** של מושגי היסוד הבסיסיים בפרדיגמה, כך שדימוי המושג אצל מתחילים של מושגים אלה, מתאים להגדרות המדויקות שלהם.

עשינו שימוש במודל זה ובהנחיות אלה על מנת לפתח סדרה של פעילויות, הכוללות משימות שונות, שיישמנו בכיתה כאמצעי ללמידה מטעויות. נתאר כמה מהן.

דוגמאות ללימוד באמצעות טעויות

במחקר הקודם שלנו [16], הבחנו בקשיים עם כמה היבטי OOP מיוחדים, ביניהם ההיבטים אודות כיתוב, יצירה וזימון של שיטות. אנו מדגימים את יישום המודל של למידה באמצעות טעויות בשלוש דוגמאות. כל אחת מהדוגמאות היא חלק מפעילות בהיבט אחד או כמה היבטים בנושא. כל אחת מהדוגמאות מוצגת בהתאם למרכיבים שהוצגו לעיל.

א. חלק של הפעילות בנושא קשיים עם כיתוב (כתובת לעומת תוכן) -

1. תפיסה שגויה: השוואה בין שמות העצמים היא השוואה בין התכונות שלהם.

2. הגדרה דקלרטיבית נכונה: השוואה בין שמות העצמים היא השוואה בין כתובות העצמים ולא בין תכני העצמים.

הגדרת הפער(*): הצרה של המושג עצם בגלל חוסר ההתייחסות אל שם העצם, כאל משתנה ששומר את כתובתו של העצם בזיכרון.

3. פעילות כיתה: כל סטודנט קיבל משימה בדף עבודה של ניתוח קוד כתוב ונדרש להתייחס להיגד המצורף תוך הסבר תשובתו לעצמו. לאחר מכן, נערך דיון בכיתה אודות התשובה השגויה ("equal") והתשובה הנכונה ("not equal") כולל הסברים.

משימה שתוביל להרחבת התפיסה:

בהינתן הקוד הבא:

```
public class Point {
    private int x_coord;
    private int y_coord;
    public Point(int x,int y) {
        x_coord = x;
        y_coord = y;
    }
}
public class ComparePoints {
    public static void main() {
        Point p1=new Point(1,2);
        Point p2=new Point(1,2);
        if (p1==p2)
           System.out.println("equal");
        else
           System.out.println("not equal");
    }
}
```

והיגד: "הפלט של התוכנית יהיה equal".

האם ההיגד נכון?

(*) בין התפיסה השגויה להגדרה הדקלרטיבית הנכונה

4. קונפליקט קוגניטיבי: הפלט "equal" שגוי, למרות שערכי תכונות העצמים זהים.

5. שיוך השגיאה: ההוראה p1==p2 משווה בין הערכים (הכתובות) של p1 ו-p2. ההשוואה היא לא בין תכני העצמים, המוצבעים על-ידי p1 ו-p2.

6. רפלקציה: הסטודנטים הבינו שהסיבה לתפיסה השגויה הזאת היא כנראה תפיסה מטושטשת אודות מודל המחשב והמנגנון של כתובת העצם. ההגדרה הדקלרטיבית של כתובת העצם מוצגת, תוך הדגשה ששם העצם שומר את כתובת העצם ולא את תכונות העצם.

עידון התפיסה: גישה לעצם קשורה לשני אלמנטים: כתובת העצם ותכונות העצם. הכתובת נשמרת במשתנה, שהוא שם העצם והתכונות נמצאות במיקום כלשהו בזיכרון המחשב, אשר מוצבע על-ידי שם העצם.

מטרתה של הדוגמה לחדד את ההבדלים בין כתובת העצם לעצם "עצמו" ולתקן את ההצרה של המושג עצם.

ב. חלק של הפעילות בנושא קשיים עם הבנת המושג יצירה של עצם -

1. תפיסה שגויה: ניתן להגדיר שם עצם ולהשתמש בו ללא צורך ליצור אותו תחילה על ידי ההוראה new ולתת לו ערכים התחלתיים באמצעות הבנאי.

2. הגדרה דקלרטיבית נכונה: תהליך יצירת העצם כולל הגדרה של שם העצם, יצירה של שטח זיכרון, ששומר את תכונות העצם וביצוע גוף הבנאי שמאתחל את תכונות העצם.

הגדרת הפער: הצרה של המושג יצירה של עצם בשל ההתייחסות לתהליך הקצאת שטח הזיכרון בעת יצירת עצם בדומה לתהליך הקצאת הזיכרון המתרחש בעת הגדרת משתנה.

3. פעילות כיתה: כל סטודנט מקבל משימה בדף עבודה של קטע קוד כתוב הכולל הנחיה למציאת שגיאה בקוד. גילוי השגיאה ילווה בהסבר עצמי של הסטודנט אודות התהליך שקורה בזיכרון המחשב בעת הגדרת ויצירת העצם.

משימה שתוביל להרחבת התפיסה:

בהינתן הקוד הבא:

```
public class Time {
   private int hour; private int minute;
   public Time(int hr,int mnt) {
         hour = hr; minute = mnt;
   }
   public boolean equal(Time other) {
      if (hour==other.hour
&&minute==other.minute)
               return true;
      else return false;
   }
}
public class CompareTime {
   public static void main() {
      Time t1,t2;
      if (t1.equal(t2))
         System.out.println("equal");
      else
         System.out.println("not equal");
   }
}
```

מצא את השגיאה בתוכנית.

4. קונפליקט קוגניטיבי: לא ניתן להשוות את ערכי העצמים כי הם לא נוצרו ואין להם ערכים. סטודנטים שיש להם תפיסה שגויה אודות הבנת תהליך יצירת העצם לא ימצאו את השגיאה שחסרה יצירה של שני העצמים תוך אתחול ערכי התכונות בעזרת הבנאי.

5. שיוך השגיאה: דימוי מושג שגוי אודות תהליך יצירת העצם. הגדרת שמות העצמים אינה יוצרת את העצמים בזיכרון.

6. רפלקציה: הסטודנטים הבינו שלאחר הגדרת שמות העצמים, יש ליצור את העצמים ולאתחל אותם בערכים בעזרת הבנאי ורק לאחר מכן ניתן להשוות בין העצמים כי יש להם ערכים בתכונות, שאותם ניתן להשוות. מושם דגש על תפקיד הבנאי ותהליך יצירת העצם.

עידון התפיסה: יצירת עצם מצריכה גם הגדרה של שם העצם וגם יצירה של שטח זיכרון, ששומר את תכונות העצם. בנוסף, תכונות העצם מאותחלות על ידי הבנאי.

מטרתה של הדוגמה לתקן את ההצרה של תפיסת המושג יצירת עצם, שאם העצמים מוגדרים אז הם נוצרו ויש להם ערכים ולכן ניתן להשוות ביניהם.

ג. חלק של הפעילות בנושא קשיים בגישה לתכונות עם הרשאת גישה פרטית -

1. תפיסה שגויה: ניתן לגשת לתכונות המחלקה, ממחלקה אחרת מבלי להתחשב בדרך שבה התכונות מוגדרות.

2. הגדרה דקלרטיבית נכונה: לא ניתן לגשת, ממחלקה חיצונית, לתכונות של מחלקה שהוגדרו עם הרשאת גישה פרטית (private).

הגדרת הפער: הרחבה של המושג זימון תכונות מופע לכך שתכונות עם הרשאת גישה פרטית מוכרות בכל שיטות המחלקה וגם בשיטות שהוגדרו מחוץ למחלקה.

3. פעילות כיתה: כל סטודנט מקבל משימה בדף עבודה של קטע קוד כתוב הכולל הנחיה למציאת שגיאה בקוד. קטע הקוד כולל היגד או אמירה שגויה. נערך דיון כיתתי בהתייחס לתשובות הסטודנטים.

משימה שתוביל להצרת התפיסה:

בהינתן הקוד הבא:

```
public class Time {
    private int hour;
    private int minute;
    public Time(int hr,int mnt) {
        hour = hr; minute = mnt;
    }
}
public class SetTime {
    public static void main() {
        Time t = new Time(12,30);
        t.hour = 11;
        System.out.println(t.hour);
    }
}
```

מצא את השגיאה בקוד הנתון.

על מנת למצוא את השגיאה ניתן להתייחס להיגד הבא: "האם נכון לומר כי: הקוד יעבור קומפילציה וירוץ בהצלחה. הפלט יהיה 11".

4. קונפליקט קוגניטיבי: הפלט "11" שגוי, למרות שערך התכונה הוא 11.

5. שיוך השגיאה: התפיסה השגויה היא ההתעלמות מהרשאת הגישה הפרטית של התכונה.

6. רפלקציה: הסטודנטים הבינו שלא ניתן לגשת אל תכונות מחלקה עם הרשאת גישה פרטית ממחלקה חיצונית. הצורה הדקלרטיבית של תכונות המחלקה מוצגות, תוך הדגשה על ההבדלים בין המושגים *private* ו-*public*.

עידון התפיסה: חידוד עיקרון הכימוס של הפרדיגמה, המתייחס למחלקה כאל קופסה סגורה שלא ניתן לגשת ממחלקות אחרות אל תכונותיה.

מטרתה של הדוגמה לחדד את תכונת הכימוס (encapsulation) של מחלקה, שהיא אחד מהמאפיינים החשובים של תכנות מונחה עצמים.

3. תיאור המחקר

אנו מציגים להלן את המתודולוגיה של המחקר ההשוואתי שלנו ולאחר מכן נציג את תוצאותיו. בנוסף, אנו מציגים את נקודת הראות של התלמידים אודות הגישה ללמידה באמצעות טעויות.

- <u>אוכלוסיית המחקר</u>

יישמנו את הגישה שלנו בכיתה בת 52 סטודנטים, בקורס תכנות ראשון שלהם בגישה המונחית עצמים, בשפת Java. המחקר שערכנו היה מחקר השוואתי, שבו ההישגים של סטודנטים אלו הושוו עם קבוצה אחרת בת 47 סטודנטים שלמדו את אותם חומרי הלימוד, אבל בדרך המסורתית ולא בדרך של לימוד מטעויות. התלמידים בשתי הקבוצות היו סטודנטים להנדסת תעשיה וניהול (לתואר B.Sc) בהתמחות מערכות מידע, במכללה להנדסה. קבוצת הניסוי וקבוצת הביקורת היו דומות במונחים של יכולות ורקע הן במדעי-המחשב והן במתמטיקה.

- <u>כלי המחקר</u>

שלושה כלים היו בשימוש: הפעילויות של הלמידה מטעויות, שאלון השוואתי על מושגי OOP ושאלון עמדות על גישת למידה מטעויות.

<u>פעילויות למידה למטרות טעויות</u>: פיתחנו שש פעילויות שונות ומגוונות. כל פעילות התמקדה בהיבטים בסיסיים של מונחי OOP ובהם: כיתוב, יצירה, תחום הגדרה, ערכי ברירת מחדל, הבדל בין אובייקט למחלקה, מאפייני הוראת

ההשמה וזימון שיטות. בכל פעילות יש כמה חלקים כאשר כל חלק בנוי לפי המודל ללימוד באמצעות טעויות, כפי שהצגנו קודם לכן. בסעיף קודם הצגנו בשלוש הדוגמאות, חלק מפעילות מלאה. כל פעילות נמשכה שיעור עד שניים. כל פעילות טיפלה במישרין בתפיסות שגויות מסוימות ובעקיפין בתפיסות שגויות נוספות. הפעילויות נבנו כך שהסטודנטים נדרשו לאתר, לזהות, לנתח ולתקן שגיאות בפתרונות שנכתבו, תוך התייחסות למאפיינים הקוגניטיביים ולמוטיבים במדעי המחשב. הפעילויות הדגישו במיוחד הגדרות מדויקות והבנת המנגנון של המודל מחשב.

שאלון השוואתי: השאלון הוצג לסטודנטים של שתי הקבוצות, לפני אמצע הקורס, וכלל שלוש שאלות, כל אחת מסוג אחר. שאלה 1 דרשה הבנת קוד; שאלה 2 דרשה כתיבת קוד; שאלת 3 דרשה השלמת קוד (שילוב של הבנת קוד וכתיבת קוד).

שאלון עמדות: השאלון הוצג לתלמידי קבוצת הניסוי לאחר הניסוי ושאל מה דעתם על היבטים שונים של גישת הלמידה מטעויות בלימודיהם.

- מהלך המחקר

קבוצת הביקורת: ההוראה של הקורס כללה הרצאה פרונטאלית ותרגול של פתרון שאלות הקשורות לנושא ההרצאה. במרבית השיעורים הוצגה לכיתה בעיה תכנותית, התקיים דיון ברעיונות ובגישות לפתרונה ואז הוצג פתרון נכון, תוך מתן דגשים על נושא השיעור.

קבוצת הניסוי: ההוראה של הקורס נעשתה על ידי המרצה הקיים. המורה קיבל חומרי למידה, שפותחו בגישה של לימוד מטעויות על ידי החוקרת וכן הנחיות לאופן הוראתם בשישה מפגשים. קבוצת הניסוי למדה את אותם נושאים ובאותו סדר של קבוצת הביקורת, אולם בשישה שיעורים (שאינם בהכרח רצופים), שכללו שלוש שעות הרצאה ושעתיים תרגול, הוצגו דוגמאות של שאלות שהצגנו בשלב ראשון של המחקר, שלב החלוקה לקטגוריות של טעויות [16]. השיעורים נלמדו בכיתה (ולא במעבדת מחשבים) אם כי בחלק מהפעילויות היה שימוש במחשב על מנת להציג ולחדד לסטודנטים טעויות מסוימות. **חשוב לציין שכמות השיעורים של שתי הקבוצות הייתה זהה. ששת השיעורים נשזרו בשיעורים ה"רגילים". כלומר, הגישה של למידה מטעויות לא האטה את קצב התקדמות ההוראה של קבוצת הניסוי**. רק אמצעי ההוראה היה שונה. לאחר מתן שאלון השוואתי, ראיינו כ-10% מהתלמידים משתי הקבוצות על תשובותיהם. נציג סיכום של הממצאים שלנו בהמשך.

- דרכי ההפעלה

דרכי ההפעלה, המופיעות בחומר הלימוד, אפשרו לסטודנטים להתנסות בסגנונות למידה שונים ומגוונים. **הפעילויות השונות עם טעויות** כללו: הצגת הגדרות שגויות, הצגת משפטי תוכנית שגויים, הצגת היגדים שגויים והסברים עצמיים שלהם, הצגת שאלות פתוחות שסטודנטים צפויים לענות עליהן בצורה שגויה והצגת פתרון שגוי של תלמיד ופתרון לדוגמה של מורה. **האמצעים השונים בשימוש הפעילויות** כללו: שימוש בדף עבודה, עבודה בזוגות, ניהול דיונים יזומים ודיונים ספונטאניים בשגיאות אופייניות ושימוש ב-case study. **האופנים השונים של חשיפת הטעויות** היו: התלמיד חושף את הטעות ומנמק מדוע זאת טעות והתלמיד חוקר את הטעות. **הדרכים השונות לתיקון הטעויות** היו: התלמיד מציע את הפתרון הנכון או שהמורה מציג את הפתרון הנכון.

4. תוצאות

ניתחנו את תשובות התלמידים בכתב ובעל פה (ראיונות). כפי שציינו קודם לכן, ההתמקדות שלנו הייתה בתפיסות מוטעות הכוללות הרחבה והצרה של מאפייני OOP בסיסיים שזיהינו במחקר קודם שלנו [16]. את התפיסות השגויות ניתן היה לראות בקטעי קוד שהתלמיד כתב, כמו גם בתשובות לשאלות הדורשות הבנה של קוד והשלמת קוד. בתשובות ניתן לזהות תפיסות שגויות או מעורפלות של הגדרות מושגי יסוד בסיסיים, מנגנונים של הפרדיגמה המונחית עצמים ומנגנון של מודל המחשב. לאחר הלימוד, התלמידים של קבוצת הניסוי הפגינו פחות תפיסות שגויות מאלו של קבוצת הביקורת. אנו מציגים להלן סיכום סטטיסטי של חלק מהתפיסות השגויות של התלמידים בשתי הקבוצות בשתי הקטגוריות השונות:

הרחבה: התייחסות לריבוי ישויות במקום לישות אחת כנדרש או הרחבת המאפיינים של המושגים השונים באופן שגוי	**קבוצת הביקורת**	**קבוצת הניסוי**
ניתן לגשת, ממחלקה חיצונית, לתכונה שהוגדרה עם הרשאת גישה פרטית	12%	4%
זימון של שיטה מעדכנת מתבצע על ידי הוראת השמה	15%	6%
ניתן לזמן שיטת מופע ללא הפעלתה על עצם	15%	0%
ניתן לייצר עצמים בעזרת בנאי ברירת מחדל, גם אם הוגדר בנאי אחר במחלקה	19%	9%
אם מאתחלים את ערכי התכונות של העצם-סימן שהעצם קיים	10%	4%
אין צורך לייצר עצמים לפני השימוש בהם	21%	9%
השמה של עצם לעצם אחר מייצרת עצם חדש	25%	18%

טבלה 1 : השוואה בין אחוז השוגים בקבוצת הביקורת לעומת קבוצת הניסוי בקטגוריה של הרחבה

הצרה: התייחסות לישות אחת במקום לריבוי של ישויות כנדרש או הצרת המאפיינים של המושגים השונים באופן שגוי	**קבוצת הביקורת**	**קבוצת הניסוי**
השמה בין עצמים מעדכנת את ערכי התכונות של העצם ולא דורסת אותו	27%	19%
"יצירה של עצם" שקולה להגדרת משתנה	29%	19%
השוואה בין שמות העצמים היא השוואה בין כל התכונות שלהם	34%	26%
העברת אובייקט מפעיל כפרמטר	29%	23%

טבלה 2 : השוואה בין אחוז השוגים בקבוצת הביקורת לעומת קבוצת הניסוי בקטגוריה של הצרה

בתשובות הסטודנטים לשאלה 2 בשאלון ההשוואתי בו מוגדרת מחלקה בשם Date המייצגת תאריך ולה 3 תכונות: יום, חודש ושנה, נתבקשו הסטודנטים להגדיר 2 תאריכים date1,date2 עם הערכים 3/5/2011 ו- 4/2/2010.

יאיר מקבוצת הביקורת לא זימן את הבנאי ורק כתב את ההגדרה הבאה Date d1={3,5,2011}. בראיון שלאחר מכן הוא הסביר: "אני אתחלתי את האובייקט d1 באופן דומה לאתחול מערך". דבר זה מבטא תפיסה שגויה של יצירת אובייקט ואנלוגיה לא נכונה לרעיון של משימה בתכנות פרוצדורלי.

לעומתו, מאיה, מקבוצת הניסוי שכתבה קוד נכון בתשובתה לשאלה אמרה: "זה לא מספיק רק להצהיר, אתה צריך ליצור את האובייקט ... הבנתי מהפעילויות למידה באמצעות טעויות שלנו שיש להכריז על האובייקט, ליצור אותו ולאתחל אותו באמצעות הבנאי ... ".

אנו מציגים להלן סיכום של תשובות של תלמידי קבוצת הניסוי לחלק מהשאלות בשאלון העמדות. בטבלה שלהלן, "1" מייצג בהחלט לא מסכים, "2"- אינני מסכים, "3"- מסכים ו- "4"- מסכים מאוד.

4 – מסכים מאוד, 3 – מסכים, 2 – אינני מסכים, 1 – בהחלט לא מסכים.	4	3	2	1
1. שימוש בדפי עבודה אישיים תרם לי.	24	26	2	
2. פעילות בזוגות שבה כל אחד חיפש טעויות אצל האחר תרמה להבנה שלי.	5	12	21	14
3. דיון במליאה אודות פתרונות שגויים שלי ושל חבריי לכיתה תרמו לי.	32	12	8	
4. התמודדות עם זיהוי טעויות בתוכנית נתונה, ציון השגיאה והצעת דרך לתיקונה תרמו לי.	23	25	4	
5. הפעילות שבה נתבקשתי להחליט האם ההיגד שגוי או לא, להסביר מה שגוי בו (אם הוא שגוי) ולנסח היגד נכון תרמו לי.	19	22	10	1
6. לימוד באמצעות טעויות עורר אתגר אינטלקטואלי וחשיבה מקורית.	33	16	3	
7. לימוד באמצעות טעויות מתאים לי יותר מדרך ההוראה ה"רגילה".	18	24	7	3
8. הייתי רוצה ללמוד גם מקצועות אחרים באמצעות לימוד מטעויות.	29	11	8	4

טבלה 3 : שאלון העמדות שניתן לקבוצת הניסוי בתום תהליך הלימוד מטעויות

בנוסף, שאלנו את הסטודנטים מספר שאלות פתוחות. בחרנו להציג חלק מתשובותיהם (התשובות נכתבו כלשונן) : "לפעמים הבנתי שאני לא יודע דברים שאני חושב שאני יודע.", "ההסבר (לעצמי) היה מאוד תורם, בגלל שאני צריך להסביר לעצמי מה שגוי ... בין דברים דומים ..." , "הפעילות עזרה לי לזכור שלא לחזור על אותה הטעות.", "זה היה כיף ... אהבתי את הדרך החדשה לעומת הדרך השגרתית של הלמידה".

5. דיון

שגיאות עלולות לעורר קונפליקטים קוגניטיביים, אשר בתורם, יניבו תהליך של התבוננות וחשיבה ביקורתית [1]. תהליך כזה עשוי לשמש לפיתוח ידע קוהרנטי (עקבי). פיתחנו והצגנו בסעיפים קודמים גישה, הכוללת פעילויות שמניבות תהליך כזה. הגישה שלנו מבוססת על ידע קודם, שהוא מעורפל או שגוי בחלקו. היא נועדה לתקן את התפיסות השגויות ולהפחית את העמימות של הידע הדקלרטיבי-הצהרתי. אנו רואים בגישה שלנו, גישה שמצמצמת פערי ידע בפרדיגמה המונחית עצמים. הגישה משלבת מאפיינים קונסטרוקטיביסטיים.

הקונסטרוקטיביזם בא לידי ביטוי בפעילויות שלנו, במובן זה שכל פעילות מניחה שיש ידע קודם והיא מתמקדת רק בתפיסה אחת מעורפלת ולעיתים שתיים. הפעילויות כוללות הסברים עצמיים, דיונים בזוגות, נימוקים ורפלקציה. אלמנטים אלו מעמיקים ומחזקים את הבנת הלומד [8]. כך גם הדיונים בכיתה המשווים בין השגוי לנכון. בדיונים כאלה אפשר להבין טוב יותר את הנכון ולהיות מודעים יותר לשגוי. יתר על כן, דיונים וחשיבה אודות הסיבות האפשרויות לטעויות (כפי שצוין בחלקי הפעילויות שהראינו בסעיף 2) מגבירים את היכולת המטה-קוגניטיבית.

חלק מהסטודנטים ציינו כי הם הבינו את התפיסות השגויות הקודמות שלהם, שלהן הם כלל לא היו מודעים. הם העלו השערות אודות המקורות של התפיסות השגויות שלהם. מקורות אלו דומים למחקרים קודמים שנערכו אודות המאפיינים של קשיים של מתחילים, וביניהם אלה של Du Boulay [3] ו-Rath & Brown [15]. Du Boulay חשף שימוש שגוי של מתחילים באנלוגיות (analogies), מטפורות (metaphors) והטלה (projection) מישות אחת לשנייה, כמו גם הכללה (overgeneralization). כל המאפיינים הללו היו קיימים גם במחקר שלנו. סטודנטים הכירו בנטיות שלהם ליישום התכנות הפרוצדוראלי בתכנות מונחה עצמים. דוגמה אחת כזאת היא תפיסות הסטודנטים אודות זימון של שיטה ללא צורך ברישום שם העצם לפניה. Rath & Brown מתארים את נטיות המתחילים לייחס יכולת של "חשיבה עצמאית" (Independent thinking) למחשב. גם במחקר שלנו, חלק מהתלמידים, הפגינו תפיסה פשטנית של תהליך יצירת העצם, מאמינים שהמחשב

"ידאג לטפל" בכל הפעולות הקשורות ליצירת העצם, מבלי שיהיה צורך לומר לו זאת מפורשות (ולזמן את הבנאי).

ממצאי המחקר שלנו מעידים כי תלמידי קבוצת הניסוי ביטאו מיומנות והבנה גבוהות יותר מתלמידי קבוצת הביקורת בפתרון השאלות, שהופיעו בשאלון ההשוואתי. חלק מתלמידי קבוצת הניסוי, טענו שהם חיזקו והעמיקו את ההבנה שלהם באמצעות פעילויות הלמידה. הם היו מסוגלים לציין את המושגים והמנגנונים באופן ברור ומדויק יותר. התוצאות המוצגות בסעיף קודם מראות את הרלוונטית ואת הפוטנציאל של הגישה שלנו בהוראה.

התוצאות מראות כי לתלמידי קבוצת הניסוי פחות תפיסות שגויות מאשר תלמידי קבוצת הביקורת והם אהבו את הפעילויות של הלמידה מטעויות. לכן, אחד מהיתרונות של הפעילויות שפיתחנו הוא העלאת המוטיבציה של הסטודנטים. מרבית הסטודנטים ראו בדרך ההוראה של לימוד מטעויות דרך המעוררת חשיבה מקורית ואתגר אינטלקטואלי. הסטודנטים הרגישו שבאמצעות הפעילויות של לימוד באמצעות טעויות הם מבינים את החומר בצורה טובה וברורה יותר ולכן מרביתם היו רוצים ללמוד גם מקצועות אחרים בדרך ההוראה של לימוד באמצעות טעויות.

נראה כי תלמידי הניסוי עידנו באמצעות הפעילויות את דימוי המושג שלהם אודות מושגים ומונחים בסיסיים ב-OOP. Vinner [18] גילה נטיות של סטודנטים לפנות לדימוי המושג במקום להגדרת המושג, בפתרון בעיות מתמטיות. דימוי המושג עשוי להיות תואם את הגדרת המושג לגמרי או באופן חלקי, או אף נוגד אותה. תפקיד ההגדרה ביצירת דימוי המושג אצל הלומדים אינו ברור: ההגדרה יכולה ליצור בתודעת הלומדים דימוי שלם של מושג או דימוי חלקי שלו, או לא ליצור דבר. הוא שם לב שחלק מהלומדים מאמצים אסוציאציות שונות מתוך הדוגמאות, שחלקן מתאימות אבל חלקן לא. הדגש בלימוד מטעויות אצלנו על ההגדרות הדקלרטיביות-הצהרתיות סייע לסטודנטים לשנות ולתקן אסוציאציות לא רצויות ולחדד את דימויי המושג של הסטודנטים.

לסיכום, אנו מציעים כמה **הצעות למורים**, על סמך הניסיון שלנו:

1. **יש לזהות את התפיסות השגויות של התלמידים שלכם.**
2. **התייחסו אל התפיסות הללו בגישה קונסטרוקטיביסטית בעת ניסיון להתמודד עימן.**
3. **פתחו פעילויות, בדומה לאלו שהצגנו והדגמנו (בסעיף 2).**
4. **הטמיעו את הפעילויות במקומות נבחרים בהוראת יסודות מדעי המחשב.**
5. **שלבו בפעילויות הסברים עצמיים, עבודה בזוגות ודיון בכיתה.**
6. **הדיונים בכיתה יהיו כרוכים בתהליך של ייחוס שגיאה (error attribution) ורפלקציה (reflection) על הפתרונות השגויים והנכונים, ופנייה להגדרות מדויקות של מושגים.**
7. **הדיון יסתיים עם מפרט הצהרתי מדויק של המונחים והמושגים הנלמדים.**
8. **לאורך כל הפעילות והדיון, יש לעניין את התלמידים ולגרום להם להרגיש שהם גם לומדים וגם נהנים.**

מורה שירצה ללמד באמצעות טעויות יוכל להתבסס על המודל שלנו ללימוד מטעויות תוך שימוש בפעילויות ובחלקי הפעילויות שפיתחנו. אנו תקווה שהממצאים יעזרו למורים להכיר בצורה טובה יותר את הקשיים של התלמידים שלהם ויאפשרו להם להתמודד איתם ביתר קלות. טעויות מספקות לנו, המורים, מידע על תפיסות של לומדים. כדאי לנסות לאתרן ולפנות לתיקונן תוך התבססות על הידע הקיים אצל תלמידים, בבחינת – "חנוך לנער על-פי דרכו" (משלי, פרק כ"ב, פס' 6).

8. מקורות

[1] Borasi R. (1996). *Reconceiving Mathematics Instruction: A Focus on Errors*, Ablex Pub.

[2] Confrey, J. (1990). What constructivism implies for teaching. *Journal for Research in Mathematics Education. Monograph Vol 4*, NCTM Inc, 107–122.

[3] Du Boulay, B. (1986). Some Difficulties of Learning to Program, *Journal of Educational Computing Research*, 2(1), 57-73.

[4] Fleury, A. E. (2000). Programming in Java students-constructed rules. *SIGCSE Bulletin*, *32*(1), 197-201.

[5] Garner, S., Haden, P. & Robins, A. (2005). My program is correct but it doesn't run: a preliminary investigation of novice programmers' problems. *7th Australasian conf. on Computer Education*, p.173-180.

[6] Ginat, D. (2003). The greedy trap and learning from mistakes, *Proc of SIGCSE 34*, ACM Press, 11-15.

[7] Ginat, D. (2008). Learning from wrong and creative algorithm design, *Proc of SIGCSE 39*, ACM Press, 26-30.

[8] Grobe C. S. & Renkl A. (2007). Finding and fixing errors in worked examples: can this foster learning outcomes?, *Learning and Instruction*, 17, 612-634.

[9] Holland, S., Griffiths, R. & Woodman, M. (1997). Avoiding object misconceptions. *ACM SIGCSE Bulletin*, *29*(1), 131-134.

[10] Melis, E. (2004). Erroneous examples as a source of learning mathematics, *Proc of CELDA 2004*, 311-318.

[11] Melis, E., Sander, A., & Tsovaltzi, D. (2010). How to support meta-cognitive skills for finding and correcting errors, *Proc of the AAAI Fall 2010 Symposium*, 64-68.

[12] Ohlsson, S. (1996). Learning from performance errors, *Psychological Review* 103:241–262.

[13] Pinkerton, K. D. (2005). Learning from errors, *The Physics Teacher*, *43* (8), 510-513.

[14] Ragonis, N. & Ben-Ari, M. (2005). A long-term investigation of the comprehension of OOP concepts by novices, *Computer Science Education*, 15(3), 203-221.

[15] Rath, A. and Brown, D.E. (1995). Conceptions of human-computer interaction: A model for understanding student errors, *Journal of. Educational Computing Research*, Vol. 12(4), 395-409.

[16] Shmallo, R., Ragonis, N. & Ginat, D. (2012). Fuzzy OOP: expanded and reduced term interpretations, *Proc of ITiCSE 12*, ACM Press, 309-314.

[17] Siegler, R. (2002). Microgenetic studies of self-explanation, in Granott, N. & Parziale, J. (eds.), *Microdevelopment, Transition Processes in Development and Learning*, Cambridge University Press, 31–58.

[18] Vinner, S. (1983). Concept definition, concept image and the notion of function, *Intl Journal of Mathematics Education in Science and Technology*, 14, 293-305.

[19] Yerushalmi, E. & Polingher, C. (2006). Guiding students to learn from mistakes, *Physics Education*, *41*, 532-538.

על הורשה מרובה והורשה בכלל

אודי מלכה

מכללת אורט ביאליק

המאמר מתאר את הדרך שעברתי עם התלמידים ממטלה בה התבקשו התלמידים ליישם בשפת התכנות C++ תרגיל בנושא הורשה מרובה ועד הרהורים ותובנות שעלו בכיתה מדיונים וחקר אודות מנגנון ההורשה בכלל והאלטרנטיבות שלו בשפות Java ו- C#.

ממטלה תמימה להרהורים על הורשה

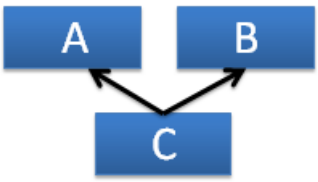
בסיום הסמסטר החלטתי לתת לתלמידים מטלת סיכום אודות מספר נושאים אשר נלמדו במהלך הסמסטר. במטלה התבקשו התלמידים לבחור תחום כלשהו מתוך מגוון של תחומים ומתוכו לבחור נושא, רעיון או אפילו סיטואציה ולייצג אותה באמצעות אובייקטים והיחסים ביניהם ממחלקות שיבחרו לבנות על פי שיקול דעתם.

הדגשתי בפני התלמידים שאחת הדרישות החשובות היא ההורשה המרובה שכן תרגלנו אותה לא מעט במעבדה והיא מנגנון שאינו קיים בשפה המונחית עצמים השנייה שלמדנו והיא Java.

מה זו בכלל הורשה מרובה?

בשפת התכנות C++ מחלקה רשאית לרשת יותר ממחלקה אחת.

נניח שקיימת מחלקת A וקיימת מחלקת B. מחלקת C יכולה לרשת גם את A וגם את B:



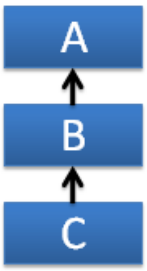

```
class A{...};
class B{...};
class C: public A, public B{...};
```

זה מאפשר למחלקת C לקבל מאפיינים שקיימים במחלקות A ו- B מבלי שלאלו (מחלקות A ו- B) יהיו יחסי הורשה ביניהן.

שאר המצבים של הורשה הקיימים בשפת java ו- C# קיימים אף הם ב- C++:

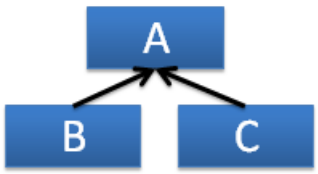
1. מחלקה יכולה לרשת מחלקה אחרת שאף היא (המחלקה האחרת) ירשה מחלקה כלשהי:

```
class A{...};
class B: public A{...};
class C: public B{...};
```



המשמעות היא שמחלקת C ירשה את מחלקת B שירשה את מחלקת A.

2. שתי מחלקות יכולות לרשת את אותה המחלקה:




```
class A{...};
class B: public A{...};
class C: public A{...};
```

המשמעות היא שהן מחלקת B והן מחלקת C ירשו את מחלקת A.

יחד עם זאת, בכל מה שקשור לשפת C++ המאמר מתמקד בהורשה המרובה בלבד.

הרעיון של הגדרת תחומים מהם יבחרו התלמידים את הדוגמא למימוש נועד למנוע מהם לבחור דוגמאות קרובות לאלו שניתנו בכיתה ושניתן למצוא בספרי לימוד, אך גם כדי לספק להם חומר למחשבה כאשר התחומים והפניות אליהם היו בנושאי איכות הסביבה, אנרגיה, רובוטיקה, ננו-טכנולוגיה, ביולוגיה, רפואה, גנטיקה וחלל.

תוך זמן קצר קיבלתי מצד התלמידים תלונות אודות הקושי למצוא דוגמאות להורשה מרובה ועל כן הסכמתי שחלק מזמן המעבדה יוקצה עבור עבודות אלו. הקושי של העבודה היה כבר בשלב הראשון ולמעשה רק בו – למצוא דוגמא מתאימה להורשה מרובה. בכיתה התפתח דיון כיתתי (מבלי שתכננתי אותו) ובו נזרקו לאוויר דוגמאות אפשריות ותלמידים שונים העירו מדוע לדעתם הדוגמא מתאימה או לא. מתוך הדיון הזה, הרעיון של הורשה מרובה זוקק לידי תובנה שמדובר במחלקה שיש בה מאפיינים של שתיים או יותר מחלקות אשר ייתכן מצב שביניהן אין שום דבר משותף. עד לרגע הזה הרעיון הזה היה ברור לכולם אך התלמידים מעולם לא התעמקו בו כי המשימות שהוטלו על ידי עד כה היו לבנות מחלקות שאופיינו מראש ועל כן הם לא נדרשו לחשוב בעצמם על מחלקות שעונות על ההגדרה הזו.

ועכשיו, כמורים למדעי המחשב - קחו לכם מספר דקות ונסו לחשוב בעצמכם על דוגמא טובה להורשה מרובה. מה אתם מכירים ויכולים להעלות בדעתכם שהוא סוג של שני "דברים" שאין ביניהם קשר? מסתבר שבאמת מאד קשה למצוא דוגמאות לבד. לכל דוגמא שניתנה בכיתה שכבר התלהבנו ממנה וחשבנו שהנה מצאנו דוגמא טובה להורשה מרובה - פתאום גילינו שיש בה משהו שמפר את ההתאמה שלה.

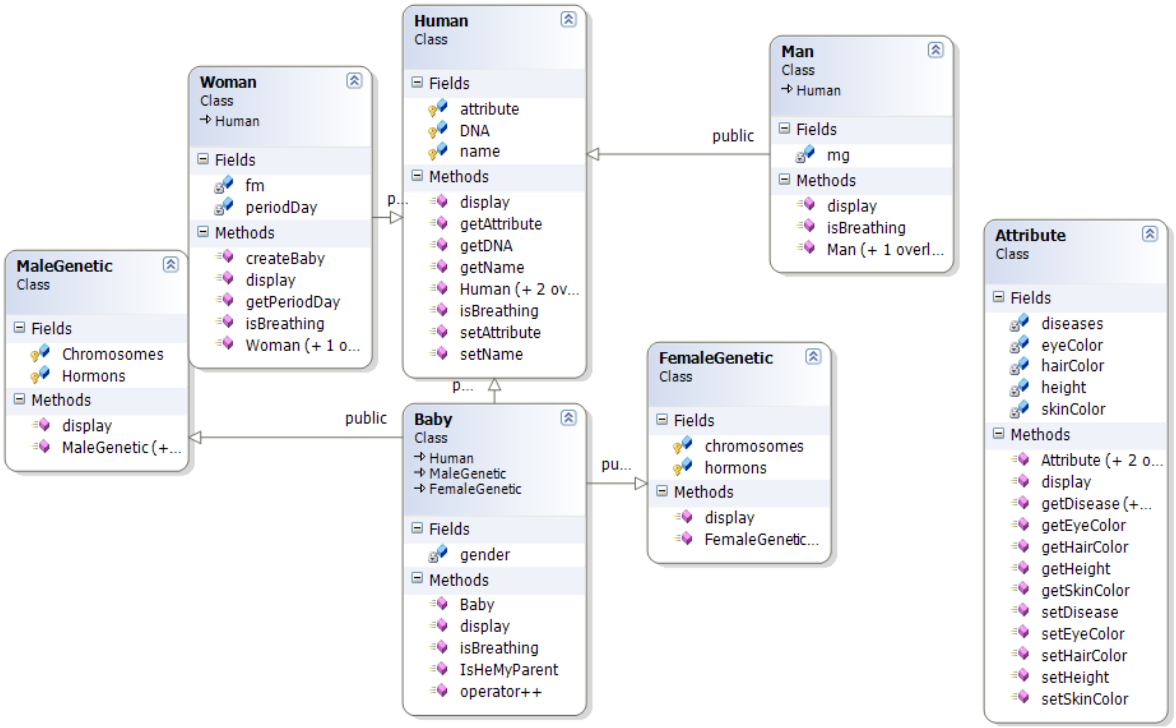
תלמיד אחד חשב על הרעיון הפשוט מתחום הביולוגיה: אם הנושא הוא הורשה אזי נמציא מחלקות המייצגות הורים (אם ואב בנפרד) ומחלקת תינוק תירש את שתיהן. מיד עלתה השאלה מה המאפיינים השונים שיש לאב ולאם והאם הם מצדיקים בנייה של מחלקה נפרדת עבור כל אחד מהם. בסופו של דבר התלמיד הוסיף עוד שתי מחלקות המייצגות את המטען הגנטי של הגבר ושל האישה והמאפיין הייחודי של האישה היה תאריך המחזור. למחלקת גבר ואישה היה, בנוסף, מאפיין קבוע המייצג את הכרומוזומים שלהם וכן מערך של תווים המייצג את ההורמונים. למחלקת אישה הייתה פעולה שמטרתה הייתה להחזיר אובייקט מסוג תינוק (שכן עד לאחרונה האם נשאה את העובר ברחמה ולא הייתה לגבר אפשרות להיות פונדקאי) והיא קיבלה כפרמטר אובייקט מסוג גבר. חלק מהפעולות שבנה התלמיד שימשו לבדיקות סטטיסטיות למחלות עבור העובר על פי המטען הגנטי של הוריו. השאלה העיקרית שעלתה בעקבות הדיון בדוגמא הזו היא האם המטען הגנטי של גבר ואשה הוא מאפיין ייחודי המצריך מחלקה נפרדת או שמא מדובר בערכים שונים של DNA, ומכיוון שבשניהם ה- DNA הוא אנושי, על כן אין צורך בבניה של מחלקה נפרדת. (ראו תרשים מחלקות של ההורשה הביולוגית בעמוד הבא)

תלמיד אחר הציג יחסים בין אובייקטים מתחום האסטרונומיה. הוא הגדיר מספר מחלקות בסיס אשר המרכזית שבהם מייצגת גרם שמיים (Astronomical object) שהוא עצם טבעי משמעותי אשר מצוי בחלל. מלבדה, הוגדרו גם מחלקות המייצגות תנועה של גרם השמיים וכן מחלקה המייצגת מסלול (תנועת לווין סביב גרם שמיים אחר). מחלקת כוכב ירשה את מחלקת גרם שמיים ומחלקת פלנטה ירשה את מחלקת כוכב וגם את המחלקות של תנועה ומסלול. בנוסף, הוגדרה מחלקה המייצגת כוכב לכת ננסי (Dwarf planet) שהוא סוג של גרם שמיים ונמצא בתנועה, אך בהגדרתו הוא אינו לווין (ולכן כוכב לכת ננסי לא ירש את מחלקת מסלול). הרעיון המרכזי שעלה מהדיון בדוגמא זו הוא שמתוך הגדרת הדמיון והשוני עולים מאפיינים ומצבים אשר יכולים להיות מאופיינים באמצעות מחלקות חדשות שלא חשבנו עליהן קודם לכן.

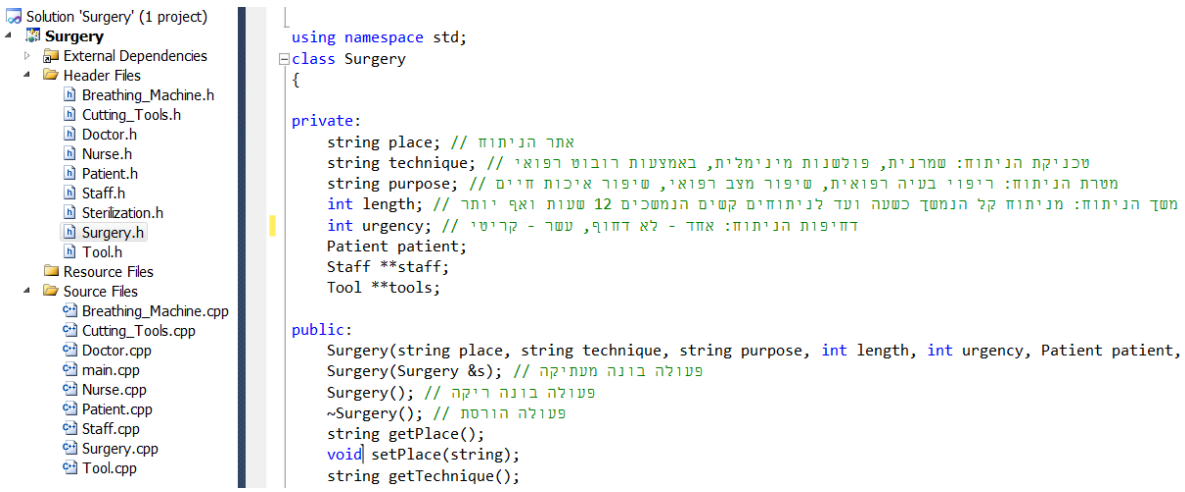
תלמיד נוסף בחר את תחום הרפואה ואפיין סביבה של חדר ניתוח ואת התנהלות הניתוח עצמו. הוא בנה מחלקה המייצגת איש צוות ושאותה ירשו מחלקת רופא ומחלקת אחות. לרופא הוא הגדיר שתי אפשרויות של התמחות: מנתח או מרדים. בנוסף, התלמיד הגדיר מחלקה המייצגת כלי עבודה (לשימוש

הרופא לצורך הניתוח או ההרדמה), מחלקת מכונת הנשמה אשר ירשה את מחלקת כלי עבודה ומחלקת כלי חיתוך אשר ירשה את מחלקת כלי עבודה ואת מחלקה נוספת אשר מייצגת פעולות סטרילזיה. כמו כן הוא בנה מחלקת פציינט ומחלקת ניתוח אשר כללה מאפיינים שונים של הניתוח.

מחלקת סטרילזיה הייתה אבסטרקטית עם פעולה אחת ומתוך הדיון בכיתה עלה כי ראוי היה שתיהפך לממשק (interface) אם המימוש היה בשפת תכנות המספקת תמיכה בממשקים כמו java ו- C#.

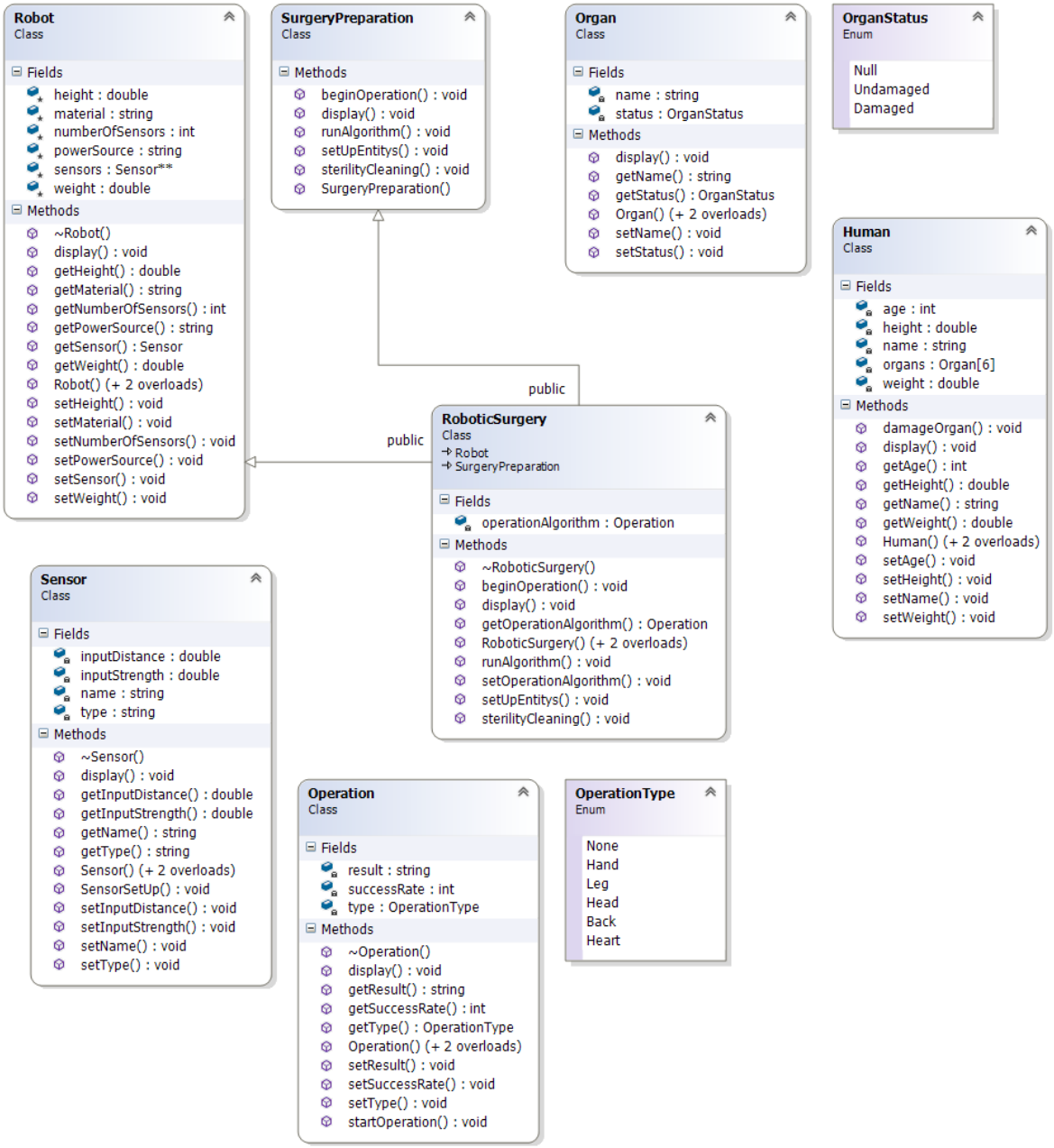


תרשים מחלקות של ההורשה הביולוגית



תמונה של מחלקת ניתוח מתוך סביבת הפיתוח Microsoft Visual C++ 2010

תלמיד אחר בחר בתחום הרובוטיקה ואפיין אף הוא מצב של ניתוח אך באמצעות רובוט. לשם כך הוא אפיין מחלקות המייצגות איברים, מצב איבר, בן אדם, סנסורים (של הרובוט) ניתוח וסוג הניתוח. ההורשה המרובה התבטאה בהורשה של שתי מחלקות שאותן בנה התלמיד: מחלקת רובוט ומחלקת הכנה לניתוח. שני מחלקות אלו נורשו על ידי מחלקת ניתוח. השאלה העיקרית שעלתה בעקבות הדיון בדוגמא הזו היא האם יש צורך במחלקה המייצגת הכנה לניתוח או שניתן למזג את הפעולות שלה (שכן היא מורכבת רק מפעולות) עם מחלקת ניתוח.



תרשים מחלקות של ניתוח באמצעות רובוט

מבלי שתכננתי, הדיונים בכיתה היו עבורי כמורה משמעותיים הרבה יותר לצורכי הוראה מההגשות של העבודות שהכינו התלמידים. הכיתה הפכה לחדר דיונים במחלקת פיתוח של חברת הייטק. הניסיון של התלמידים להצדיק את הדוגמא שלהם ולהבליט את נכונותה מול ההערות של תלמידים אחרים שהסבירו מדוע יש כאן סטייה מן הרעיון של הורשה מרובה או מדוע הדוגמא אינה מייצגת באופן מובהק את השימוש הנכון בהורשה מרובה - הניסיון הזה הפך לדבר העיקרי והיה עקרוני ופורה יותר מקוד שעובד בצורה תקינה או באופן יעיל. מעבר לקושי למצוא דוגמאות טובות, הדוגמא הראשונה שהוצגה במאמר העוסקת בירושה של תינוק את תכונות אביו ואימו הבליטו (לאחר שהחל התלמיד בשלב המימוש) בעיה נוספת שלא עלתה קודם לכן בכיתה והיא נלמדה בנוסף מעבר לחומר הלימוד שאותו תכננתי ללמד. הבעיה הזו נקראת diamond problem ונרחיב עליה בסעיף הבא.

הורשה מרובה ו- diamond problem

נתחיל בדוגמא. נניח שיש לנו מחלקות המייצגות התקן USB והתקן רשת תקשורת. ב- C++ מחלקת מתאם רשת יכולה, כאמור, לרשת את שתי המחלקות הנ"ל:

```
class USBDevice {...};
class NetworkDevice {...};
class WirelessAdaptor: public USBDevice,
public NetworkDevice {...};
```

נניח שלשתי מחלקות הבסיס יש מאפיין שנקרא serial המייצגת את המספר הסידורי של המכשיר ופעולה שנקראת getSerial (אשר מחזירה את ערכו של serial). כעת נתבונן בקוד:

```
WirelessAdaptor adaptor(5442, 181742);
cout << "Serial is: "<<adaptor.getSerial();
```

איזו גרסה של הפעולה getSerial תופעל? של מחלקת USBDevice או של מחלקת NetworkDevice ?

הפתרון לבעיה זו קיים באמצעות צורת כתיבה המדגישה את הגרסה של המחלקה הרצויה אך זו עדיין דרך מסורבלת לפתור בעיה שנוצרה בעקבות הורשה מרובה.

הפתרון:

```
WirelessAdaptor adaptor(5442, 181742);
cout << "Serial is:
"<<adaptor.USBDevice::getSerial();
```

בעיה נוספת שנוצרה בעקבות הורשה מרובה היא בעיה הנקראת diamond problem. נתבונן במחלקות הבאות המייצגות התקן, סורק, מדפסת ומדפסת משולבת:

```
class PoweredDevice{...};
class Scanner: public PoweredDevice{...};
class Printer: public PoweredDevice{...};
class Copier: public Scanner, public
Printer{...};
```

מדפסת משולבת יורשת את מדפסת ואת הסורק. שני אלו יורשים את מחלקת התקן. האם מדפסת משולבת יורשת פעמיים את מחלקת התקן? ניתן לפתור בעיה זו באמצעות הגדרת הירושה של מחלקת התקן כווירטואלית, אך גם כאן ניתן לראות סרבול שההורשה המרובה יוצרת ושלא היה קיים קודם לכן.

הפתרון:

```
class PoweredDevice{...};
class Scanner: virtual public
PoweredDevice{...};
class Printer: virtual public
PoweredDevice{...};
class Copier: public Scanner, public
Printer{...};
```

בעיות עם הורשה בכלל

החקירה של התלמידים ושלי אודות בעיות של הורשה מרובה הביאה אותנו לגלות כי יש לא מעט מאמרים שעוסקים בבעיות הקשורות להורשה בכלל ושאינן ספציפיות להורשה מרובה וקיימות גם בשפות java ו- C#. חיפוש במנוע החיפוש google של הביטוי "composition vs inheritance" יעלה לא מעט מאמרים בנושא.

כדי להבהיר את הבעיה עם הורשה באופן כללי נחזור לעקרונות של תכנות מונחה עצמים: כימוס, הורשה ופולימורפיזם. נתמקד בשני העקרונות הראשונים.

אובייקטים הם ישויות עצמאיות עם מצב (state) ופונציונליות (methods). העקרון של כימוס מאפשר שחלק מהמצב ומהפעולות יהיו פרטיים (private). באופן כזה, מתכנתים המשתמשים בקוד שנכתב על ידי מישהו אחר אינם נדרשים להכיר את אופן המימוש, בוודאי לא זה הפרטי. והיתרון שהוא אולי החשוב ביותר – שינוי של חלקים של מתכנת אחר שהוגדרו כפרטיים אינו אמור להפריע לשאר חלקי המערכת.

הורשה היא מנגנון שנועד להסדיר את הדמיון והשוני בין אובייקטים שונים ונעשה בה שימוש כאשר נקבע שמחלקה אחת היא סוג של מחלקה שנייה. הבעיה היא שכימוס והורשה, ששניהם עקרונות של תכנות מונחה עצמים, לעיתים אינם משלימים אחד את השני אלא דווקא סותרים ומעוותים אחד את השני. הורשה פוגעת בכימוס (Vlissides, Helm, Johnson & Gamma, 1995).

הפגיעה באה לידי ביטוי בשתי נקודות מהותיות:

1. שדות או פעולות שהוגדרו כ- protected במחלקת הבסיס נגישות לכל מי שירש את המחלקה בה הם נמצאים וכמי שכתב את מחלקת הבסיס אין לי שליטה מה יעשו בהם.

2. בהורשה קיים קשר מתמיד (coupling) בין מחלקת הבסיס למחלקה הנגזרת ושינויים במחלקת הבסיס (גם בשדות ופעולות המוגדרות כ- private) עלולים להשפיע על המחלקה הנגזרת (ripple effect).

כדי להבליט את הבעייתיות בהורשה נבחן שתי דוגמאות. נניח שיש לנו מחלקה המייצגת לוח דו-מימדי של משחק. יש במחלקה מאפיינים המייצגים את הקואורדינטות (X ו- Y) וכן את הלוח עצמו (מערך דו-מימדי). כעת החלטנו שאנו מעוניינים להפוך את המשחק לתלת-מימד ולשם כך בנינו מחלקה חדשה אשר ירשה את זו שבנינו קודם לכן. הוספנו למחלקה היורשת מאפיין לייצוג מימד נוסף (גובה - בשם Z). אז מה הבעיה? כל הפעולות של חישוב מרחקים ממחלקת הדו-מימד אינן נכונות וגם לא ניתן להשתמש בהן (כיוון שחישובי מרחקים בתלת-מימד שונים מאלו של דו-מימד). אם כך, לשם מה ירשנו? מקרים שבהם אנו נדרשים לדרוס פעולות במחלקת נגזרת אבל איננו משתמשים בפעולות של מחלקת הבסיס צריכים להעלות אצלנו נורה אדומה בקשר להתאמה של הורשה לפתרון הבעיה. דוגמא נוספת: נניח שבנינו מחלקת מלבן בעלת המאפיינים אורך ורוחב. ריבוע במתמטיקה הוא מקרה פרטי של מלבן ועל כן בנינו מחלקה המייצגת ריבוע אשר ירשה את מחלקת מלבן. בפעולות של עדכון אורך ורוחב, לא שכחנו לממש במחלקת ריבוע שעדכון אורך כלשהו יעדכן לאותו הערך גם את הרוחב ולהיפך (בריבוע כזכור האורך והרוחב שווים). עתה נניח כי מתכנת אחר התבקש לבנות פעולה המקבלת כפרמטר מלבן ומשנה את האורך והרוחב שלו. במידה והפעולה תקבל אובייקט מסוג ריבוע, היא בפועל תשנה את האורך והרוחב שלו ולאחר מכן תשנה אותם שוב לערך אחר. שתי הדוגמאות הללו סותרות את מה שקרוי "Liskov substitution principle". עקרון ההחלפה או התחלופה של ברברה ליסקוב טוען שבכל מקום בתוכנית שבו יש אובייקט מסוג אחד אזי החלפתו באובייקט ממחלקה נגזרת צריכה לספק משמעות זהה בהתאם לציפיות או כפי שהיא ניסחה זאת ליסקוב במקור:

Let $q(x)$ be a property provable about objects x of type T.
Then $q(y)$ should be provable for objects y of type S where S is a subtype of T.

העיקרון הזה הוא העיקרון השלישי מתוך חמישה עקרונות של תכנון מונחה עצמים שנקראים* SOLID.

כדאי לזכור שהיתרון של הורשה הוא קיצור כתיבה ותמיכה בפולימורפיזם. אם בעקבות ההורשה אנו נדרשים לחזור אל מחלקת הבסיס ולשנות שם קוד או שאין לנו שימוש בקוד שכתבנו אז פספסנו משהו. גם התמיכה בפולימורפיזם ניתנת לפתרון שאינו מחייב שימוש בהורשה.

(*) SOLID נוצר על ידי ראשי התיבות של **S**ingle responsibility, **O**pen-closed, **L**iskov substitution, **I**nterface segregation and **D**ependency inversion.

מה האלטרנטיבה להורשה?

הביטוי "composition vs inheritance" שהוצג קודם לכן מבטא את שתי הגישות: א) הורשה (inheritance) מייצגת יחסי "Is a" ונראה שהשימוש בה טבעי כאשר אנו רוצים ליצור אובייקט שהוא "מאותו הסוג" של אובייקט ממחלקה שכבר קיימת אך משודרג יותר. לדוגמא: מכונית היא סוג של רכב, בן אדם הוא סוג של יונק. ב) הכלה (composition) מייצגת יחסי "Has a" והשימוש בה נראה טבעי כאשר אובייקט אחד נדרש להכיל אובייקט אחר. לדוגמא: למכונית יש מנוע, לבן אדם יש כתובת מגורים. כדאי לציין כי לעיתים המציאות הרבה יותר מורכבת וקיימים מצבים בהם, למשל, אדם מתפקד באותה תקופת זמן כמהנדס בפרויקט אחד אך הוא מנהל צוות בפרויקט אחר או עוזר הוראה אשר מתפקד כאיש צוות של האוניברסיטה אך הוא גם סטודנט (Arnold, Gosling & Holmes ,1996). הבחירה בפתרון כזה או אחר היא תלוית הקשר וצורך. למשל, אם אובייקטים מסוג עובד ומנהל נדרשים לשנות מאפיינים שלהם כבני אדם אז יש לשקול פתרון שמבוסס על הורשה ובו מחלקת עובד ומחלקת מנהל אשר ירשו את מחלקת בן אדם. המבחן לשימוש בירושה או הכלה הוא ההקשר ולא אם האובייקטים הנדרשים מייצגים במציאות יחסי "סוג של" או "מכיל את".

הרושם שנוצר אצלי, מתוך מספר רב של מאמרים שקראתי, הוא שיש כיום נטייה למעט בשימוש בהורשה ולבחור יותר בכיוון של פתרון ההכלה. ההכלה מאפשרת הצהרה של הפניות (reference) במחלקה ויצירה של אובייקטים בזמן ריצה כאשר יש בהם צורך. מסיבה זו שינוי של מחלקה אחת אינו מצריך הידור של מחלקות אחרות. לעומת זאת, בגישת ההורשה יצירה של אובייקט מאתחלת את כל המאפיינים שלו, כולל אלו שירש, ושינוי של מחלקת הבסיס מצריך הידור נוסף גם של המחלקות היורשות. שילוב ממשקים (interfaces) יחד עם ההכלה מספק גם את יתרונות הפולימורפיזם.

ממשקים הם התחליף ב- Java ו- C# למנגנון ההורשה המרובה שנעדר מהם. בהקשר לפתרון השימוש בממשקים כדאי לציין כי בדומה למניעה להשתמש בהורשה כאשר אין לנו צורך לרשת מאפיינים מסוימים חשוב לזכור כי אין לרשת ממשק שיש בו פעולות שהמחלקה לא נדרשת אליהן. עקרון זה נקרא "Interface segregation principle" (העיקרון הרביעי ב- SOLID) והוא מדגיש את הצורך בפיצול ממשק אחד כללי למספר ממשקים כאשר מחלקה נדרשת לממש רק חלק מהפעולות שהיו בממשק הכללי. יש ספרים בהם חלק מהתרגול הוא להסב תוכניות המשתמשות במנגנון ההורשה לפתרון ההכלה ולבחון את היתרונות והחסרונות של כל מנגנון בהקשר של התרגיל (Dietel ,2009). המעבר מהורשה להכלה צובר תאוצה והמושג "composition over inheritance" אשר מייצג את הטכניקה להשגת מאפיינים של פולימורפיזם ומחזור קוד באמצעות הכלה הופך להיות נפוץ יותר ויותר.

השימוש בהכלה נראה פחות טבעי לעיתים מהורשה, אולי בגלל צורת החשיבה שהורגלנו אליה לאחר שלמדנו ותרגלנו בעצמינו הורשה, אך כדאי להכיר את חסרונות ההורשה ואת הפתרונות האלטרנטיביים הקיימים לה.

מקורות

Arnold, K., Gosling, J., & Holmes, D. (1996). *The Java programming language*(Vol. 2). Reading: Addison-wesley.

Dietel, P. (2009). *Java how to program*. PHI.

Vlissides, J., Helm, R., Johnson, R., & Gamma, E. (1995). Design patterns: Elements of reusable object-oriented software. *Reading: Addison-Wesley*, *49*, 120.

אתר האינטרנט של המרכז הארצי

בחודשים ספטמבר 2014 – דצמבר 2014 קיימנו סקר של אתר האינטרנט של מרכז המורים הארצי למדעי המחשב. להלן מספר תוצאות של הסקר.

החומרים שנבדקו במסגרת הסקר

1 חומרי עזר למודול סייבר ואינטרנט (חט"ב)

2 מחקר שטח של המשימה המסכמת במודול גיליון אלקטרוני (חט"ב)

3 שילוב המעבדה בהוראת מדעי המחשב

4 יסודות: חומרי עזר למורה (חומרים לתכנית הלימודים החדשה שפיתחו מורי הניסוי בשנת תש"ע)

5 יסודות: הוראת עצמים ומחלקות, מערך של עצמים, ניתוח של שאלות בגרות (חומרים שפיתחו משתתפי קורס מורים מובילים תשע"ד)

6 יסודות: מאגר שאלות (חומרים שפיתחו משתתפי קורס שאלות מבחן תשע"ג)

7 פרויקטים שניתן לשלב בהוראת יסודות (חומרים שפיתחו משתתפי קורס מורים מובילים תש"ע)

8 יחידה שלישית 899127: מאגר שאלות (חומרים שפיתחו משתתפי קורס שאלות מבחן תשע"ג)

9 עיצוב תכנה: מאגר שאלות

10 מודלים חישוביים: מאגר שאלות

11 הצעות, רעיונות וחומרים לשיווק מדעי המחשב

12 סקר ספרות מעולם המחקר בהוראת מדעי המחשב

13 יסודות: מדריך למורה

14 עיצוב תכנה: מדריך למורה

15 תכנות מונחה עצמים: מצגות

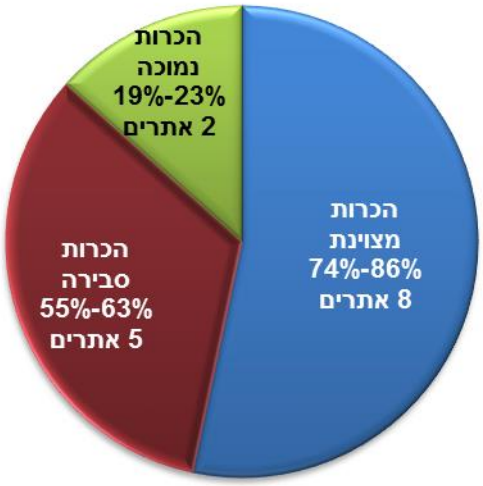
ניתוח תוצאות הסקר

מתוך 15 אתרים שנבדקו: 8 אתרים מוכרים על-ידי מספר גדול של מורים (74%-86%), 5 אתרים נוספים מוכרים על-ידי יותר מ- 50% מהמורים (55%-63%), ורק 2 אתרים מוכרים על-ידי מספר נמוך יותר (19%-23%) של מורים.

אתר	אני מכיר	אני משתמש (מבין המורים שמכירים את האתר)
1	23%	64%
2	19%	67%
3	76%	84%
4	86%	81%
5	74%	88%
6	85%	88%
7	55%	46%
8	63%	69%
9	83%	87%
10	62%	78%
11	59%	61%
12	57%	58%
13	74%	77%
14	76%	84%
15	74%	63%

שלושת האתרים המוכרים ביותר למורים שענו על הסקר (83%-86%) הם שני אתרים שמוקדשים לשאלות (אתר 6 שאלות ל"יסודות" ואתר 9 שאלות ליחידה הרביעית) ואתר (4) הכולל חומרי עזר לתכנית הלימודים החדשה ביסודות שפותחו על-ידי המורים שהשתתפו בניסוי של הוראת התכנית החדשה בשנת תש"ע. החומרים שנמצאים באתרים האלה הם בעלי אחוזי שימוש גבוהים ביותר וחלק גדול מאד מהתגובות של המורים התייחסו לאתרים האלה.

לדוגמה(*), "החומרים ערוכים נכון עם הסברים ותשובות ועזרו לי גם בבחירת שאלות וגם בחיבור שאלות בהקשרים דומים"; "החומרים מאד עוזרים בתכנון מערכי שיעור ובניית מבחנים. מאגרי השאלות - חומר מצוין שעוזר בעיקר למורה חדש"; "בתור מורה חדש, הדבר שהיה בעיני השימושי ביותר הוא הרעיונות של איך ללמד" (מתייחס לאתר 4) "ואיך להנגיש את החומר לתלמידים"; "הוראת עצמים ומחלקות (מתייחס לאתר 4) - קיבלתי רעיונות מעולים לבניית מחלקות - וכן הכוונה איך לבנות את מערך השיעור".



אתר נוסף שראוי לציון הוא אתר המעבדה (3). 76% מהמורים שענו על השאלון מכירים אותו ו- 84% מהם עושים שימוש בחומרים שנמצאים באתר הזה. להלן מספר תגובות של מורים שקיבלנו, שמתייחסות לאתר המעבדה:

- "שילוב המעבדה ביסודות - מאוד מובנה ועוזר מאוד כמעבדת חקר";
- "מעבדה תחילה: החומרים ערוכים היטב ומתאימים לתלמידים. שילוב מעבדה בהוראת יסודות – גישת מעבדה תחילה מאתגר את התלמידים, הופך את השיעור למעניין ומאפשר לתלמיד להתקדם בקצב שלו";
- "שילוב המעבדה בהוראת מדעי המחשב, באתר יש חומרים מוכנים ללימוד בדרך הגילוי בצורה מדורגת עם הסברים מלווים, החומרים המוצלחים מאד לדעתי הם המעבדות בגישת המעבדה תחילה, שקיימות בכל הנושאים ומהוות עבורי גיוון ללמידה ולדעתי הן ערוכות מצוין";
- "החומרים של שילוב המעבדה תחילה בהוראת מדעי המחשב מעולים: מאפשרים לנו ללמד בצורה אחרת לאפשר לתלמיד לגלות והמורה יותר כמכוון ולא בעל הידע היחיד";
- "השתמשתי הרבה במעבדות שיש באתר. התלמידים מאוד אהבו והתקדמו מצוין. צריך להכין עוד מהסגנון הזה להמשך יחידה רביעית וחמישית";
- "החומרים השימושיים ביותר עבורי הם שילוב המעבדה בהוראת מדעי המחשב – אלו חומרים שאני משתמשת בהם בשוטף, והם מעולים";
- "אני משתמשת שנים במעבדות שמאפשרות לכל תלמיד להתקדם בקצב שלו ולמורה לקדם את הקבוצה בצורה יעילה".

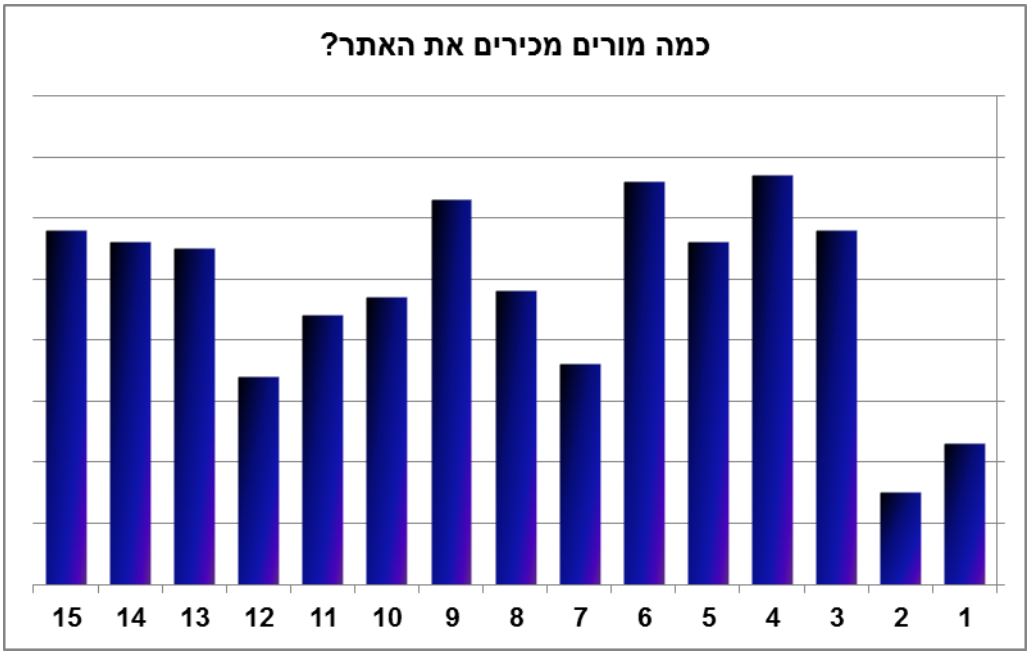
לפי ממצאי הסקר, שני אתרים (1,2) מוכרים על-ידי מספר נמוך (19%-23%) של מורים. האתרים האלה הם חומרים לחטיבת ביניים ולכן תוצאות הסקר הגיוניות כיון שרוב מורי התיכון לא זקוקים לחומרים האלה ולא משתמשים בהם.

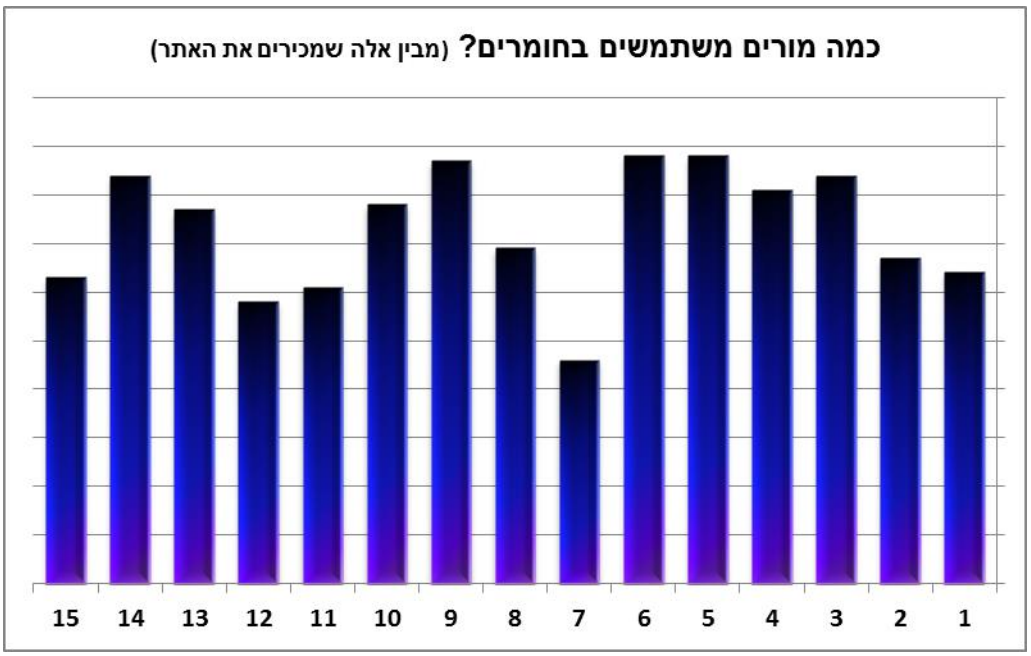
מבין כל האתרים שנבדקו, רוב המורים עושים שימוש משמעותי (61%-88%) בחומרים, ורק ביחס לשני אתרים (7, 12) יש פחות מורים שעושים שימוש בחומרים.

אתר 7 (פרויקטים ל"יסודות") מוכר ל- 55% מהמורים שענו על הסקר. יתכן שהסיבה נוגעת לעובדה שתכנית הלימודים הנוכחית לא דורשת פיתוח פרויקטים במסגרת "יסודות". יחד עם זאת, 46% מהמורים שמכירים את האתר כתבו שהם כן משתמשים בחומרים האלה. זו עובדה מעודדת שמספר גבוה יחסית של מורים משלבים פיתוח פרויקטים בתוך ההוראה, למרות שזה אינו חובה.

אתר 12 מוקדש לסקר ספרות ולכן הוא אולי נתפס כפחות רלבנטי לשימוש ישיר במהלך ההוראה. יחד עם זאת, 59% מהמורים שמכירים את האתר ציינו שהם משתמשים בו. אנחנו מעריכים שלפחות חלק מהמורים האלה ממשיכים בלימודים לתואר שני ושלישי ולכן הם יכולים להיעזר בחומרים במהלך הלימודים שלהם.

* שמות המורים שענו על הסקר שמורים במרכז הארצי





בהזדמנות זו אנחנו מודים לכל מי ששיתף פעולה וענה על הסקר.

אם לא עניתם עדיין על הסקר, תוכלו לעשות זאת בכתובת הבאה

http://cse.proj.ac.il/Tofes/Seker/Seker.asp

הטכניון – מכון טכנולוגי לישראל

קול קורא למורים למדעי המחשב בבית הספר העל יסודי

שלום רב לך המורה למדעי המחשב

אם יש באמתחתך מערך-שיעור להוראת המקצוע מדעי-המחשב בעל-יסודי, שחביב עליך במיוחד, וברצונך לשתף בו את עמיתיך או מורים אחרים למדעי-המחשב שיכולים להתברך בכך, הקול הקורא הזה מיועד אליך.

<u>כמה מילות רקע</u>:
"רמזור למורה" הוא פרויקט שיצא לדרך בשנת תשע"א במסגרת "קשר חם" – מרכז מו"פ לקידום, שיפור ורענון החינוך המתמטי בישראל המתנהל בטכניון. המטרה היא לאפשר לכלל ציבור המורים למתמטיקה ליהנות ממאגר אינטרנטי מצטבר של מערכי שיעור במתמטיקה לפי תכה"ל, בבחינת:

"עצור לקריאה ועיון, היכון להפעלה, צא לדרך בכיתתך".

המאגר מצטבר על תוכנה שפותחה במיוחד על ידי חברת אומניסול כדי לשרת את המטרה הזאת, ונקראת "תוכנת רמזור".

<u>וכעת לגופו של עניין</u>:
לאחרונה התקבל מגוגל מענק המאפשר להרחיב את המאגר המצטבר על תוכנת רמזור גם להוראת המקצוע מדעי-המחשב. כדי להעמיד מאגר ראשוני של מערכי שיעור במדעי-המחשב על התוכנה, דרושים מורים כמוך, שישמחו לשתף את עמיתיהם במערכי השיעור שלהם.

המענק מאפשר לתת תמורה של 500 ₪ לכל אחד מ-40 המערכים הראשונים שיתקבלו על גבי התבנית שתימסר למעוניינים בכך. התמורה איננה מותנית בשיפוט כלשהו של המערכים (למעט בקרת שגיאות). התמורה מותנית רק בהכנת המערך על פי ההנחיות ובהבעת נכונות לפרסם את המערך לידיעת כלל ציבור המורים למדעי-המחשב. ניתן יהיה לקבל ייעוץ והנחייה בהכנת המערך ממומחה בתחום, אך אין הכרח להשתמש באפשרות המוצעת. אין הגבלה על מספר המערכים שמותר לכל מורה להגיש.

לפרטים נוספים נא לפנות אל רחלי שחם, מזכירת "קשר-חם" בדוא"ל:
Project.ramzor@gmail.com ולציין בשורת הנושא: רמזור למדעי המחשב. הפונים יקבלו בדוא"ל חוזר הנחיות ברורות יחד עם התבנית שתאפשר להעלות את מערך השיעור שיוגש לתוכנת "רמזור".

סמינר קיץ תשע"ד למורים מובילים

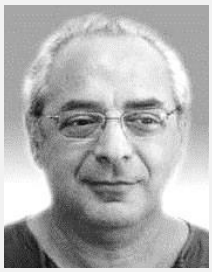
ביולי 2014 התקיים בניר עציון סמינר קיץ תשע"ד למורים מובילים במדעי המחשב, בהשתתפות 25 מורים.
ההרצאות שניתנו בסמינר:

תכנות בעזרת מיליוני דוגמאות: חיפוש קוד לשימוש חוזר בעזרת מערכת חדשנית לכריית מידע

פרופ' ערן יהב, הפקולטה למדעי המחשב, הטכניון

תכנות מודרני כולל שימוש נרחב בממשקים (APIs), וכתיבת קוד אשר מחבר ביניהם. בעולם שבו יש מיליוני דוגמאות קוד זמינות, מישהו במקום כלשהו אולי כבר כתב את הקוד שהמתכנתת צריכה, או קוד שהיא יכולה ללמוד ממנו. האתגר הוא למצוא את הקוד הזה. בהרצאה תואר מנוע לחיפוש קוד באופן סמנטי — חיפוש ש-"מבין" מהם רצפי הפעולות שהקוד מבצע, ואיך ניתן למצוא קוד שמבצע פעולות דומות. תוארה מערכת פורמלית לכרייה אוטומטית של מפרטים טמפורליים מתוך קטעי קוד. המערכת עושה שימוש בניתוח סטטי של קוד, על מנת לבנות ייצוג מבוסס אוטומטים של התנהגות הקוד. בהרצאה תוארו רעיונות בסיסיים בניתוח סטטי של קוד, ניתוח סטטי למטרות כריית מפרטים, ניתוח סטטי של כמויות קוד גדולות, ואלגוריתמי למידה, אשר משלבים את הידע מתוך קטעי הקוד השונים.
ההרצאה מבוססת על עבודה משותפת עם אלון משנה, שרון שוהם, והונגסוק יאנג.

משחקים, אלגוריתמים, מתמטיקה ומה שביניהם: הרפתקה אלגוריתמית במחוזות חידאתו, סודוקו ומספרים גדולים



ד"ר גיורא בנדק, סיסקו ישראל; ממציא משחקים, ביניהם: "חידאתו"

"חידאתו" הינו משחק לוגי, המפתח את החשיבה וההיגיון. הוא נפוץ בכל העולם, בעיתונים, ספרים ועוד. במקומות רבים הוא משמש ללימוד חשיבה לוגית-מתמטית. בהרצאה הוצגו המשחק חידאתו, איך פותרים פאזלים ללא ניחושים, בעיה חישובית מעניינת הקשורה לחידאתו, הצעה לפרויקט לתלמידים מתקדמים המערב אלגוריתמיקה, עיבוד מבוזר (ענן) ועוד. כמו כן, הוצגו משחקים נוספים ומספרים גדולים הקשורים אליהם, דיון על היכולות שלנו לתפוס מספרים, על הקושי לתפוס מספרים גדולים מאוד, ומה אפשר לעשות בנידון.

דימות וצבע: תורת הצבע, עיבוד תמונה וטכנולוגיה

פרופ' חגית הל-אור, החוג למדעי המחשב, אוניברסיטת חיפה

צבע, במשמעותו הפיזיקלית, הוא תופעה רב ממדית, אך מרחב הצבע הנתפס ע"י בני אדם, ומרחב הצבע המשמש ביישומים טכנולוגיים, הינו תלת-ממדי (למשל: מרחב ה-RGB כבסיס למצלמות ומסכים, או מרחב ה-YMC למדפסות).
בהרצאה הוצגו היסודות של תורת הצבע בהיבטיו הפיסיקליים, ותרגומם להיבטים התפיסתיים והטכנולוגים. הוצגו מספר בעיות ופתרונות, כפי שבאים לידי ביטוי בטכנולוגיות השונות, כגון: תיקון צבע (white balancing) והשלמת צבע (demosaicing) במצלמות דיגיטלית, מיפוי צבעים ממסך למדפסת, דחיסת תמונת צבע ועוד.

להיות כלואים בזמן: על דחיקתם והדחקתם של החיים הממשיים בעולם הווירטואלי

ד"ר סיגל נאור-פרלמן, ארגון "דרך רוח" לקידום מדעי הרוח בישראל

ההרצאה עסקה באשליית הקיום בזמנים מקבילים וירטואליים, הדוחקים את הזמן הממשי של הקיום הפיזי של האדם בעולם, כמו גם את השאלות הגדולות על משמעות החיים, ועל הבחירה כיצד לעצב משמעות זו.

מאז נכנס האינטרנט לחיינו במידה הולכת וגוברת בשני העשורים האחרונים, חל שינוי של ממש בתפיסת האדם את עצמו, את מעמדו בעולם, ואת הקשר שלו עם בני אדם אחרים בכל רחבי העולם. כתוצאה מכך, "הזמן האנושי", כפי שהכרנו אותו, השתנה באופן רדיקלי. כיום יכול האדם לשבת בחדר ממשי על כיסא ממשי, ובה בעת להתקיים במקביל בכמה יקומים וירטואליים, הכוללים תיבות דואר אלקטרוני, מנועי חיפוש, רשתות חברתיות שונות וצ'טים מהסוגים השונים. יתרה מזאת, ההתקדמות הטכנולוגית בעיצוב האפשרויות והאפליקציות של הטלפונים הניידים, אף מאפשרים לאדם להתקיים ביקומים וירטואליים אלה תוך תנועה ומעבר תדיר ממקום למקום.

ואולם, מדובר רק באשליה של קיום בו זמני ביקומים הווירטואליים הללו, משום שהאדם עצמו, גופו הממשי, יכול להימצא רק במקום אחד. במלים אחרות: אותה חלוקה שהייתה ברורה בעבר, בין תודעת האדם ובין זמנו הממשי של הגוף, נשארה על כנה, אך האפשרות הווירטואלית להתקיים בו זמנית בכמה יקומים מקיימת את האשליה, שהאדם עצמו נע באופן ממשי מאתר אחד למשנהו. אשליה זו מכתיבה קצב מהיר הרבה יותר של התודעה, הנדרשת להגיב לגירויים המתקבלים תדיר, וכן ציפייה לתגובות מהירות לגירויים הנשלחים. כך מוצא את עצמו האדם כשהוא כלוא בין זמן עבר (זמן הגירוי שנשלח), לזמן עתיד (זמן קבלת התגובה), כשזמן ההווה, או תפיסת זמן ההווה, הולכים ומצטמצמים לטובת תודעה כוזבת זו. הקצב המהיר שעוד הולך ו-"משתכלל", המוכתב על ידי מצב חדש זה, עלול להשכיח, או לדחוק לקצה התודעה את קיומו הממשי של הגוף בעולם ואת מגבלותיו הפיזיות, אבל הרבה יותר מכך, את העובדה שיש, או שניתן, לתת את הדעת על כך: לשאול שאלות, הן את שאלות החיים הגדולות הנוגעות לכאב ולעונג, לאושר ולסבל, לחיים בתוך הקשר חברתי, היסטורי ותרבותי, והן כל שאלה אחרת ההופכת את האדם לאדם.

"רוח דיגיטלית": מדעי הרוח וחישוביות

ד"ר יעל נצר, המחלקה למדעי המחשב, אוניברסיטת בן-גוריון

מדעי הרוח הדיגיטליים (Digital Humanities) הם שדה מחקר, פיתוח ויצירה, העושים שימוש במחשוב ובכלים ממוחשבים בתחומי הידע השונים של מדעי הרוח — היסטוריה, ספרות, תרבות ושפה. מעבר לכך, זוהי תנועה של חידוש, העצמה והעשרה של מדעי הרוח, שהתפתחה עם מהפכת המדיה והמידע של העשורים האחרונים, ועם האפשרויות שטכנולוגיות המחשב, הדיגיטציה והאינטרנט פותחות בפני מדעי הרוח. עבור מדעי הרוח, שחומר הגלם, הכלי והתוצר המרכזי שלהם — הטקסט, עבר לפורמט הדיגיטלי, המהפכה הדיגיטלית, שמשמעה שחלק גדל והולך של חיינו מתרחש ומנוהל על המחשב ובאינטרנט, היא משמעותית לא פחות מאשר בכל מדע אחר.

בניגוד לעמדה פסיבית, המסתפקת בברכות, קינות או ניסיונות להתמודד עם התמורות שמביאה מהפכה זו, מדעי הרוח הדיגיטליים לוקחים בה חלק פעיל, ביקורתי ורפלקטיבי, תוך שהם פועלים כמשתתפים מובילים בפיתוח המשאבים, בבניית כלי המחקר וגישות המחקר הדיגיטלי, ובעיצוב קהילת הידע באקדמיה ומחוצה לה. בעשור האחרון בעיקר, נפתחו ברחבי העולם מרכזים ותכניות למדעי הרוח הדיגיטליים, והללו מאורגנים לכדי קהילה בין לאומית תוססת.

הרצאה עסקה בהתפתחות התחום וענפיו בעשורים האחרונים בעולם ובארץ דרך תחנות דרך, פרויקטים, כלים ומחקרים, והועלו כמה משאלות היסוד שלו.

אקראיות בחישוב

פרופ׳ רונן שאלתיאל, החוג למדעי המחשב, אוניברסיטת חיפה

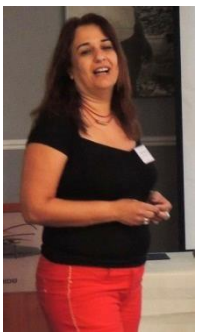
אלגוריתמים ופרוטוקולים הסתברותיים עושים שימוש בהטלות מטבע אקראיות. לגישה זו שימושים רבים באלגוריתמים וקריפטוגרפיה.

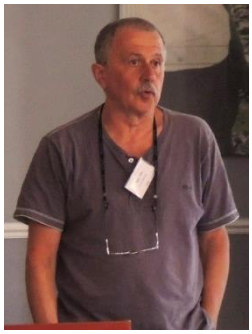
בהרצאה נסקרו שני כווני מחקר תיאורטיים המנסים להבין:

1. באילו משימות אפשר להדמות (simulate) אלגוריתמים הסתברותיים באופן יעיל ע״י אלגוריתמים דטרמיניסטיים?
2. כיצד ניתן להפיק ״אקראיות טהורה״ מדגימה מהתפלגויות ״בעלות אקראיות מסויימת״, הזמינות למחשבים?

..... החומרים שהוצגו בסמינר ניתנים לצפייה באתר המרכז הארצי .....

תמונות הסמינר המופיעות בעיתון צולמו ע״י רחלי צרניחוב וד״ר דן אהרוני





ביום השני של הסמינר, יצאו משתתפי הסמינר לסיור מודרך בזכרון יעקב ולאחר מכן התקיימה ארוחת ערב חגיגית במסעדה "קאזה ברונה". למרות שביציאה מהארוחה נשמעה האזעקה הראשונה של מבצע "צוק איתן", המשכנו בתכנית והצלחנו לקיים את הסמינר עד סופו.

..... להתראות בסמינר הבא .....

חומרי למידה שפורסמו באתר בשנת תשע"ד

אתם מוזמנים להיכנס לאתר ולהיעזר בחומרים שפותחו במהלך תשע"ד:

- **חומרי עזר לחטיבת הביניים בנושא "סייבר ואינטרנט" שפותח ע"י משתתפי קורס מורים מובילים לחטיבת הביניים תשע"ד, בייעוצם של ד"ר איליה אברבוך, מר איתי דברן, ד"ר דורון זוהר, ופרופ' עירד יבנה**

http://cse.proj.ac.il/Y14/cyber.pdf

- **חומרים שפיתחו משתתפי קורס מורים מובילים תשע"ד: הוראת עצמים ומחלקות, מערכי שיעור לנושא "מערך של עצמים", ניתוח שאלות בגרות**

http://cse.proj.ac.il/Y14/materials/index.htm

הבטים בהוראת מדעי המחשב: משוב לגליון ינואר 2015

קוראים יקרים

לאחר שסיימתם לקרוא את העיתון, אנא מלאו ושלחו משוב זה בהקדם למינהלת מל״מ.

תודה על שיתוף הפעולה,

צוות המרכז הארצי

משוב לגליון ינואר 2015 של "הבטים בהוראת מדעי המחשב"

נא למלא ולשלוח אל:

מינהלת מל"מ

בניין קנדה, קומה 1

קריית הטכניון, חיפה 32000

פקס: 04-8295010

1. שם בית הספר _____

2. שם המשיב_____ מספר המורים שעיינו בגליון זה בבי״ס _____

3. סמנו במשבצת המתאימה את חוות דעתכם:

				הערות
חוות דעת כללית על הגליון	טובה מאד	טובה	לא טובה	
החשיבות של כתב העת	רבה מאד	רבה	לא חשוב	
מידת העניין	מעניין מאד	מעניין	לא מעניין	
תרומה לעבודתי	תורם מאד	תורם	לא תורם	

4. הערות נוספות:

_____

_____

_____

_____

_____