

תבניות פדגוגיות להוראת מדעי המחשב

כתבה וערכה: ד"ר ברוריה הברמן

מבוא

תהליכי הוראה מלווים בהתמודדות עם בעיות פדגוגיות מגוונות כדוגמת יצירת מוטיבציה אצל תלמידים, ארגון ההוראה שתתמוך בלמידה משמעותית, הערכה, ועוד. תבניות הן פתרונות לבעיות החוזרות על עצמן בהקשרים שונים. בפרט, תבניות פדגוגיות מתארות בצורה מתומצתת וברורה את מומחיותם של מומחי הוראה ומורים מנוסים, ומאפשרות העברת מומחיות זו בתוך קהילת המורים. הגישה הופכת את המומחיות ל"זמינה" מאפשרת למורים לרכוש מומחיות בלי ללמוד ישירות מהמומחה (שאינו תמיד זמין). הגישה מאפשרת קיצור דרך ברכישת המומחיות בכך שניתן ללמוד מניסיונם של עמיתים ותיקים ומומחים, ואין הכרח לבנות את המומחיות האישית על סמך התנסות אישית בלבד.

חשוב לציין כי התבניות אינן מתארות רעיונות פדגוגיים חדשים לבעיות פדגוגיות ידועות, או פתרון מקורי לבעיה פדגוגית חדשה, אלא מהוות פורמליזציה והכללה של פתרונות ידועים, בעלי מאפיינים משותפים, שהוכחו כפתרונות מוצלחים לבעיות פדגוגיות שזוהו בקהילה. בהתאם, תבנית פדגוגית המנוסחת על ידי מומחה מייצגת למעשה מלבד הידע האישי שלו גם ידע של מומחים נוספים. המומחה המנסח תבנית פדגוגית למעשה "לוכד" מומחיות מוכחת, ולעתים עליו לעסוק בשאיבת ידע (בדומה לתהליך שעוברים מהנדסי ידע בפיתוח מערכות מומחות).

תבניות פדגוגיות מיועדות לשימוש חוזר וצריכות להיות נוחות להתאמה וליישום במקרים ספציפיים. הן צריכות להיות מתוארות באופן מובן כך שמורים שונים יוכלו להשתמש בהן בהקשרים ובמצבים שונים. קהילות של חוקרים ואנשי חינוך כדוגמת **פרוייקט תבניות פדגוגיות** (Pedagogical Patterns Project) עוסקים באיסוף טיפוסים שונים ומגוונים של תבניות במטרה לעזור למורים להורות ולתלמידים ללמוד (Sharp, Manns, & Eckstein, 2003).

בפרט, תבניות פדגוגיות תופסות לאחרונה מקום חשוב בהוראת מדעי המחשב. איש החינוך והחוקר Joseph Bergin פיתח תבניות פדגוגיות להוראת מדעי המחשב אותן נתאר באתר זה. **התבניות המקוריות של ברגין.**

מאמץ מיוחד הוקדש לפיתוח תבניות פדגוגיות בתחום טכנולוגיות מונחה עצמים. ראו למשל:

Eckstein, 2001; Jalloul, 2000; Sharp et al, 2003

תבניות פדגוגיות פותחו לתכנון, פיתוח והפעלה של קורסים. ראו למשל:

Bergin, 2001; Bergin 2002b; Eckstein, 2000; Eckstein et al, 2002, Eckstein et al 2003

Bergin (2002) סבור שתבניות פדגוגיות להוראת מדעי המחשב יכולות להתייחס לבעיות ברמות הפשטה שונות:

(א) תבניות המתייחסות לבעיות פדגוגיות כלליות הקשורות למטרות על של הוראת המקצוע ולמושגים החוזרים על עצמם בנושאים שונים הנלמדים בתחום;

(ב) תבניות המתייחסות לבעיות הקשורות להוראת נושא מסוים או פרדיגמה מסוימת;

(ג) תבניות המתייחסות לבעיות יותר ספציפיות, אך עדיין מתאימות למגוון בעיות. מכאן, שקהילת מפתחי התבניות מכירה בחשיבות ובצורך של פיתוח תבניות ברמות שונות של הפשטה וכלליות. יחד עם זאת קיימת ביקורת על כך שתבניות פדגוגיות מסוימות לעתים אינן כלליות מספיק ואינן מייצגות מהות של בעיה נטולת הקשר ולעתים אינן ברמת הפשטה מספקת (Fincher, 2002).

תוכן:

- מרכיבי תבנית פדגוגית
- אימוץ תרבות התבניות הפדגוגיות ליצירת קהילת מורים שיתופית מתוקשרת ומודל להעברת תרבות התבניות הפדגוגיות לקהילת המורים
- תבניות פדגוגיות להוראת מדעי המחשב של Joseph Bergin (כולל תקציר התבניות וסיכום מפורט של תבניות לדוגמה)
- תבניות פדגוגיות שהוצעו על ידי מורים למדעי המחשב: זה הקטן גדול יהיה, זרימת מידע שתי הצעות של תבניות חדשות להוראת מדעי המחשב שהוצעו על ידי מורים למדעי המחשב, בצד הצעות לשימוש בהן ובתבניות שפיתח Joseph Bergin לתכנון פעילויות למידה.
- רשימת מקורות מומלצים

דוגמה לשימוש בתבנית פדגוגית לארגון למידה סביב מושגים מרכזיים

עיין בתבנית הפדגוגית Early Bird שמציע Joseph Bergin.

הוא מדבר על רעיונות גדולים (עקרונות מרכזיים) בקורס אותו המורה מלמד.

<http://csis.pace.edu/~bergin/PedPat1.3.html>

1. ערוך רשימת רעיונות / עקרונות מרכזיים במדעי המחשב:

2. בחר 2 נושאים (יחידות לימוד) בתכנית הלימודים ותאר כיצד הרעיונות / עקרונות מרכזיים במדעי המחשב שציינת באים לידי ביטוי בנושאים אלה.

רעיון/עיקרון מרכזי במדעי המחשב		כיצד באים לידי ביטוי בכל נושא	
		נושא 1:	נושא 2:
1			
2			
3			
4			
5			

מרכיבי תבנית פדגוגית

תבניות פדגוגיות מנוסחות בצורה קומפקטית בפורמט שהתפתח במשך הזמן שתכליתו לאפשר העברת המומחיות והידע הפדגוגי בצורה פשוטה למי שזקוק לו (Bergin et al., 2001; Sharp et al., 2003). לפי Bergin ([ראו כאן](#)) לתבנית פדגוגית יש את המרכיבים ההכרחיים הבאים: תיאור הבעיה הפדגוגית, רעיון מוצע לפתרון, ההקשר בו הפתרון אמור להיות מיושם, וכוחות (forces) שיש להביא בחשבון בשיקולים ליישום הפתרון. הכוחות עשויים לעזור להצלחת יישום הפתרון הנבחר או לחילופין עלולים להפריע ולגרוע מהצלחתו. הכוחות הם למעשה אילוצים הקיימים במערכת בה עולה הבעיה הפדגוגית ויש להתחשב בהם בבחירת יישום הפתרון בהקשר המסוים בו רוצים לפתור את הבעיה. תבניות פדגוגיות אינן עומדות בפני עצמן וניתן להצביע על קשרים ביניהן. מכיוון שתבנית מהווה הכללה של פתרונות שהוכחו כמוצלחים, אחד ממרכיבי התבנית מתייחס לתיאור דוגמאות מוכרות ליישום מוצלח. מרכיב נוסף דן בתוצאות ובהשלכות אפשריות של יישום הפתרון, כמו כן שיקולים לאי שימוש בתבנית מוצגים בעת הצורך. מרכיב נוסף מתייחס למשאבים מיוחדים שיכולים לשמש ליישום הפתרון, כמו גם מקורות ספרות שעשויים להועיל.

חשוב לציין שהמבנה שמציע Bergin אינו היחיד המקובל בקהילת התבניות הפדגוגיות. הצעות נוספות ניתן למצוא ב- (Sharp et al., 2003) ובפרסומים אחרים בנושא.

מרכיבי תבנית פדגוגית על פי Bergin	
Pattern name	שם תבנית
Thumbnail	כותרת
Problem/issue	תיאור הבעיה / הנושא
Audience/context	קהל יעד / הקשר
Forces	כוחות
Solution	הצעת פתרון
Discussion/consequences/implementation	דיון/ תוצאות/ יישום
Special resources	משאבים מיוחדים
Related patterns	קשר לתבניות אחרות
Example instances (known uses)	דוגמאות מוכרות ליישום מוצלח
Contraindications	הסתייגויות
References	מקורות

תרבות התבניות הפדגוגיות

אימוץ תרבות התבניות הפדגוגיות ליצירת קהילת מורים שיתופית מתוקשרת

סעיף זה מתמצת רעיונות לאימוץ תרבות התבניות הפדגוגיות ליצירת קהילת מורים שיתופית מתוקשרת המוצגים ב- (Haberman, 2003).

תכנית הלימודים בארץ במדעי המחשב וטכנולוגיות מידע כוללת נושאים רבים ומגוונים, ובלתי אפשרי לצפות ממורה שיכיר את כל הנושאים ועל אחת כמה וכמה שיהיה מומחה בכלם. לרוב מורים מתמקצעים בנושאים מסוימים ומלמדים אותם לאורך זמן, אך לעתים הם נאלצים ללמד נושאים שאינם מנוסים בהוראתם.

בקהילת המורים יש ידע פרקטי מומחי מבוזר שלרוב נשאר נחלתם הפרטית של המורים. העברת הידע הפרקטי המומחי בהוראה אינה מתאפשרת על ידי תהליך ממושך של לימוד שוליה ממומחה כמקובל באומנויות אחרות כיוון שהמורה הוא לרוב עובד כיחיד בכיתתו, ואין אפשרות לצפייה מתמשכת בעבודת עמיתים פרט אולי לתהליך הכשרת מורים בטרם גיוסם למערכת (pre-service). רכישת ידע פדגוגי/מקצועי במהלך העבודה (in-service) מתבצעת מדי פעם, לרוב לעתים רחוקות בגלל חוסר תקציב, דרך השתלמויות מורים.

לרוב אין למורים נגישות לספרות מקצועית המתייחסת למחקרים בתחום הוראת מדעי המחשב וקשיי תלמידים בלימוד המקצוע ובתפיסת מושגים. יחד עם זאת יש להם הבנה פרקטית של הבנת קשיי תלמידים, הודות לנסיון הוראה מצטבר.

האינטראקציה בין מורים עמיתים נעשית לרוב במסגרת המוסד ומתבטאת בדרך כלל בהחלפת תרגילים ומבחנים בלבד. השאיפה היא שמורים למדעי המחשב יערכו הכרות עם הרעיון של תבנית פדגוגית והצגתה כביטוי ופורמליזציה של מומחיות וידע פרקטי מוכח ויעריכו את התבניות ככלי פדגוגי להוראת מדעי המחשב וכמנוף ליצירת תקשורת פרקטית בקהילה.

אימוץ התרבות של תבניות פדגוגיות כאמצעי תקשורת בין מורים ובפרט מורי מדעי המחשב, יכולה לשמש להחצנת ולהצננת הידע הפרקטי המומחי ויצירת מאגר תבניות פדגוגיות המייצג ידע זה. בהתאם, יש חשיבות רבה לקידום של תקשורת מקצועית בין מורים ויצירת קהילת עמיתים שיתופית. יש מקום לתקשורת בשלוש רמות: מומחה-ל-מומחה, מומחה-ל-מתחיל, מתחיל-ל-מתחיל. פיתוח התבניות עשוי לתרום גם לשיפור עבודתו של המפתח כיוון שהוא מלווה מן הסתם בניתוח אירועי למידה תוך התייחסות למטרות הוראה, רפלקציה וראייה רטרוספקטיבית.

מודל להעברת תרבות התבניות הפדגוגיות לקהילת המורים

על מנת שמורים יוכלו לעסוק בתבניות פדגוגיות יש לחשוף אותם לתרבות זו בשלבים. מומלץ שהמורים יתוודעו תחילה לרעיון בכללותו ולרקע ההיסטורי של התפתחות תרבות זו, ובהמשך יכירו מגוון תבניות פדגוגיות קיימות להוראת מדעי המחשב כדוגמת (Bergin, 2002). מומלץ שהחשיפה תיערך בשלבים הבאים:

- ניתוח והבנת מהותן של תבניות קיימות.
- יישום – שימוש בתבניות קיימות לפתרון בעיות ספציפיות תלויות הקשר.
- פיתוח של תבניות חדשות.

השאיפה היא שמורים יפתחו תבניות פדגוגיות כלליות ככל האפשר להוראת מדעי המחשב. יחד עם זאת, גם לפיתוח תבניות ברמת הפשטה נמוכה יש חשיבות לא מבוטלת היות והמורים המפתחים עוברים בכל מקרה תהליך של ניתוח מצב קיים, הגדרת בעיה, ופורמליזציה של הידע הפרקטי המומחי שלהם. תרומתו של תהליך זה, מעבר לתוצר המתקבל, שהוא גורם למורים להיות מודעים לתהליך ההוראה ולהסיק מניסיונם המצטבר על דרכי הוראה מכוונות מטרה. שימוש בתבניות פדגוגיות עשוי לעזור למורה בארגון ההוראה שלו וכן בהערכת תלמידיו.

חשוב שהמורים יתנסו בשימוש בתבניות פדגוגיות לפתרון בעיות ספציפיות תלויות הקשר. ההתנסות תכלול דיון במהות התבנית, זיהוי אירועי למידה ונושאים ששימוש בתבנית רלוונטי עבורם; תשומת לב מיוחדת צריכה להינתן מן הסתם גם לקשיי תלמידים החוזרים בהקשרים שונים. חיבור מערכי שיעור בהם מיושם הרעיון לפתרון המיוצג בתבנית הפדגוגית, כולל פיתוח פעילויות למידה ותרגילים מתאימים. לדוגמה, שילוב מושכל של תבניות אלגוריתמיות לפיתוח יכולות של פתרון בעיות (Muller, Haberman, & Averbuch, 2004).

תבניות פדגוגיות להוראת מדעי המחשב של Joseph Bergin

Bergin מציג 14 תבניות פדגוגיות להוראת מדעי המחשב. לדבריו, התבניות מהוות בסיס לפיתוח שפת תבניות (pattern language) שיכולה לעזור בפיתוח/ארגון קורסים במדעי המחשב. התבניות אינן כולן באותה רמת הפשטה. חלקן מתייחסות להיבטים כלליים של ארגון קורס, וחלקן מתייחסות לפרטים יותר ספציפיים ברמה של פעילויות יום-יומיות במפגש עם תלמידים.

פירוט התבניות נמצא ב- <http://csis.pace.edu/~bergin/PedPat1.3.html>.

באופן כללי ניתן לסכם ש-Bergin מצביע על שני פערים מרכזיים בהם צריך לטפל:

**** חשוב שתלמידים יקבלו תמונה כללית הוליסטית של הדיסציפלינה (או של נושא מסוים) בצד הבנת המאפיינים של מושגים/עקרונות/רעיונות בודדים, ולא "ילכו לאיבוד" בים של פריטים. נשאלת אם כן השאלה כיצד לארגן את הלמידה במטרה לפתח בתבונה אצל הלומד את שתי נקודות המבט- ההוליסטית בצד הפרטנית.**

**** אחת המטרות של לימוד מדעי המחשב היא לפתח אצל הלומד את היכולת להבין אופן פעולתן של מערכות תוכנה ותהליכי פיתוחן, בצד היכולת לפתח באופן עצמאי אלגוריתמים ומערכות תוכנה. לדעת Bergin קיים פער בין היכולת לפתח תוצר מורכב לבין היכולת להבין אותו- קל יותר להבין היבטים שונים של מערכת קיימת מאשר לפתח אותה מן היסוד. נשאלת אם כן השאלה כיצד לארגן את הלמידה במטרה לפתח אצל הלומד יכולות "קריאה" ו-"כתיבה".**

רעיונות לגישור על פערים אלו ניתן למצוא בתבניות הפדגוגיות ש-Bergin מציע. למשל, מומלץ שתלמידים מתחילים יעסקו ב- "קריאה לפני כתיבה" ויעבדו ב"קטן" על מערכות גדולות הבנויות בתבונה (על ידי המורה) בטרם יפתחו מוצר שלם באופן עצמאי.

עבודה זו כוללת תקצירים של מכלול התבניות הפדגוגיות ש-Bergin מציע בצד סיכום מפורט יותר של מספר תבניות מייצגות.

שפת תבניות לפיתוח קורס **ניתן למצוא כאן**

מכלול התבניות הפדגוגיות:

- **Early Bird** - ארגון הלמידה בקורס כך שהמושגים / הרעיונות / העקרונות החשובים יילמדו/יוצגו תחילה או כמה שיותר מוקדם.
- **Spiral** - ארגון הלמידה בקורס סביב הקניית כלים ופיתוח יכולות פתרון בעיות תוך הצגת תתי נושאים של הקורס בשזירה ובעידון הדרגתי במעגלים. ארגון הלמידה בדרך זו מאפשר לתלמידים לפתור בעיות משמעותיות בשלבים מוקדמים ככל האפשר.
- **Consistent Metaphor** - שימוש במטפורה להצגת נושא מורכב במטרה לאפשר לתלמידים לראות את "התמונה הגדולה" מבלי ללכת לאיבוד ב"ים הפרטים".

- Toy Box - מתן מבט רחב – תיאורטי ואפליקטיבי על ידי הפעלה באמצעות כלים אטרקטיביים מוכנים שהופכים את הלמידה לחוויה נעימה. התנסות עם כלים מוכנים ליצירת תוצר מורכב אותו התלמידים אינם יכולים לפתח מן היסוד במלואם באופן עצמאי. הכלים יכולים להיות "סגורים" או ניתנים להרחבה.
- Tool Box - תלמידים בונים ארגז כלים דינמי ומתפתח מן הבסיס, לשימוש חוזר עתידי (באותו קורס ובקורסים מתקדמים). המטרה מושגת על ידי תרגילים מרובי סעיפים הכוללים בין השאר מטלה לבנות כלי כללי לשימוש חוזר. מתאים לעבודה שיתופית בין צוותים.
- Lay of the Land - הבנה מערכתית של תוצרים מורכבים. התייחסות לתמונה הגדולה בצד פרטיה. יכולת בחינה והערכה של תוצר מורכב למרות שאין עדיין את היכולת לפתח מוצר בהיקף כזה באופן עצמאי. ההערכה כוללת איתור חסרונות ובעיות. נותן מושג על מהות תוצרים בהם עוסקים בעולם האמיתי.
- Fixer Upper - איתור ותיקון תקלות/שגיאות/בעיות פשוטות בתוצר גדול מוכן מוכן מראש. בדרך זו התלמיד עובד על מערכת מורכבת קיימת ולומד לבדוק אותה, לשפץ ולשפר אותה. (קריאה לפני כתיבה).
- Larger than Life - שינוי מערכת קיימת והתאמתה לדרישות חדשות. (קריאה לפני כתיבה).
- Student Design Sprint - התנסות בעיצוב ותיכון בשלבים תוך כדי עידון הדרגתי. המטרה לפתח הרגלים נכונים המתאימים לעיצוב מערכות מורכבות. עבודה בקבוצות בשלבים תוך שינוי הדרגתי בעיצוב בעקבות שאלות למחשבה המוצגות על ידי המורה שלא ניתן לענות עליהן כאשר העיצוב לוקה בחסר ואינו טוב דיו.
- Mistake - איתור ותיקון שגיאות על ידי יצירה מכוונת של שגיאות בהנחיית המורה במטרה להתנסות בבחינת תוצאות והשפעות השגיאה (בחינת תגובות המערכת וסביבת העבודה – הבנת הודעות שגיאה). למידה מאי-דוגמאות.
- Test Tube – בחינה עצמית של "מה יקרה אם?" – פיתוח יכולות חקר וגילוי של התנהגות מערכת. מתאים למקרים בהם אין כללים ברורים. בחינת השערות. דמיון לבחינת השערות במדעי הטבע.
- Fill in the Blanks - לימוד נושאים מורכבים על ידי בניית מרכיבים קטנים של תוצר. שילוב של קריאה וכתיבה ויחד עם זאת אפשר להקדים קריאה לכתיבה. למידה על ידי קריאה בצד עשייה. משימות של השלמת החסר.
- Gold Star - הערכה חיובית וחיזוק על ביצועים טובים. מתן ציון לשבח.
- Grade it Again Sam - מתן אפשרות לתיקון שגיאות ושיפור איכות תוצר מאפשר ללמוד משגיאות ומעודד שיפור הדרגתי.

תבנית Early Bird

כותרת

ארגן את הקורס כך שהנושאים החשובים ביותר יילמדו תחילה. למד את החומר החשוב ביותר, את הרעיונות המרכזיים ("big ideas") תחילה ולעתים קרובות. כאשר זה נראה בלתי אפשרי, למד את החומר החשוב מוקדם ככל האפשר.

בעיה/סוגייה

בקורס טיפוסי יש הרבה נושאים חשובים, לרוב קשורים ביניהם. קשה להחליט כיצד לארגן את סדר הוראת הנושאים כך שהתלמידים יתרשמו מהרעיונות המרכזיים ("big ideas") של הקורס. כאשר דוחים את הצגת הנושאים החשובים לשלב מאוחר, ומקדישים זמן רב לנושאים המקדימים, התלמידים עלולים לקבל רושם מוטעה על החשיבות היחסית של הנושאים הנלמדים. כמו כן, דחיית הצגת הנושאים החשובים לשלב מאוחר תמנע את האפשרות לחזור ולדון בהם ולתרגל אותם מדי פעם במהלך הקורס.

קהל/הקשר

לתבנית זו יש יישום רחב כמעט בכל תחום ונושא.

כוחות

תלמידים צריכים לדעת לקראת מה מוביל הקורס. הם צריכים להשתכנע שפרטים המוצגים בתחילת הקורס מתייחסים לרעיונות החשובים והמרכזיים בקורס. תלמידים צריכים לדעת מהם הרעיונות המרכזיים בכל קורס שהם לומדים. עליהם להיות מסוגלים להבחין בין אותם רעיונות מרכזיים לבין פרטים שתומכים בהם. לרוב תלמידים זוכרים יותר טוב את מה שהם לומדים בתחילת הקורס. לכך יכולות להיות השלכות חיוביות כמו גם שליליות בהקשרים מסוימים. אפשר להציג רעיונות מרכזיים לראשונה כמה שיותר מוקדם, גם אם הצגתם והטיפול בהם בשלב ראשוני אינו שלם וכוללני.

פתרון

תחילה זהה מהם הרעיונות החשובים ביותר של הקורס ומיין אותם לפי סדר חשיבותם. ארגן את הלמידה סביב רעיונות אלו. הצג את הרעיונות המרכזיים ואת הקשרים ביניהם כבר בתחילת הקורס, וטפל בהם שוב ושוב במהלך הקורס. מצא דרך כיצד ללמד את הרעיונות החשובים יותר תחילה. לדוגמא, אם תכנון חשוב יותר מתכנות, מצא דרך לעסוק בהיבטים של תכנון מוקדם ככל האפשר. אם פונקציות חשובות יותר ממבנה בקרה "אם-אז" בתכנות, למד פונקציות תחילה.

דיון/תוצאות/יישום

בדרך זו, הרעיונות החשובים ביותר בקורס או זוכים ליותר שימת לב מהמורה ומהתלמידים כיוון שחוזרים עליהם שוב ושוב במהלך הקורס. היישום של הרעיון קשה מכיוון שלרוב רק היבטים פשוטים של הרעיונות המרכזיים יכולים להיות מוצגים בשלב ראשוני בגלל מורכבותם. לעתים, אי אפשר להסתפק בהצגת כללית ראשונית בלבד בהתחלה, מה גם שבדרך כלל הרעיונות החשובים והמרכזיים נחשבים כמתקדמים. יש להקדיש מחשבה רבה כיצד לעשות את ההצגה בעידון הדרגתי תוך קשירת הרעיון לרעיונות נוספים ולתכנים הנלמדים באופן שוטף.

אם לפי דעת המורה אין אפשרות להציג רעיון מרכזי כבר בהתחלת הקורס, יש לפחות להימנע מללמד שיטות שעוללות להתנגש עם הרעיון או להקשות על הצגתו בשלב מאוחר יותר. למשל, אם מלמדים תכנות פרוצדורלי לפני תכנות מונחה עצמים יש להשתדל למנוע קושי במעבר לגישה מונחית עצמים. יש לנתח את ההשלכות של סדר ההוראה של הנושאים השונים.

משאבים מיוחדים

על המורה להשקיע מחשבה רבה וזמן בארגון הלמידה ובפיתוח חומרי למידה ייעודיים. רצוי לנהל קבוצות דיון עם עמיתים בעלי מטרה משותפת.

קשר עם תבניות דומות

Spiral – בתבנית זו יעשה שימוש יעיל ברעיונות המרכזיים בצורה מתפתחת.
Lay of the Land – בתבנית זו יעשה שימוש כדי להדגים לתלמידים דוגמא של "רעיון מרכזי בפעולה". כאשר יש שילוב של מספר רעיונות מרכזיים ניתן להדגים את הפעלתם תוך שימוש בתבנית *Larger Than Life*.
Fixer Upper – בתבנית זו יעשה שימוש כדי להציג את הרעיון המרכזי לראשונה.
אם הרעיון המרכזי הוא מורכב, ניתן להשתמש בתבנית *Toy Box* כדי להדגים את הרעיון.
רעיונות הקשורים זה לזה אפשר להציג תוך שימוש ב- *Tool Box*.

דוגמאות מוכרות ליישום מוצלח

הוראת עצמים תחילה (objects first) או מוקדם ככל האפשר. הוראת תכנון תחילה. הוראת רקורסיה לפני לולאות.

הספר *Karel the Robot* נכתב על פי עקרון זה ע"י Richard Pattis במטרה ללמד "פרוצדורות תחילה" בתכנות פרוצדורלי. *Karel++* נכתב ברוח דומה כדי ללמד "מחלקות תחילה" בתכנות מונחה עצמים.

הסתייגויות

זו עלולה להיות טעות להשתמש בתבנית זו כאשר לחומר יש דרישות קדם ברורות וחיוניות לרעיונות המרכזיים והקשר בין דרישות הקדם לרעיון המרכזי הוא עדין במיוחד, או שקשה לטפל בדרישות הקדם. יחד עם זאת, שימוש מושכל ב- *Toy Box* או ב- *Lay of the Land* יכול להועיל בכל זאת להציג את הרעיון המרכזי תחילה.

מקורות

Karel the Robot, Richards Pattis, Wiley, 1981.

Karel++, Joseph Bergin, Mark Stehlik, James Roberts, Richard Pattis, Wiley, 1997.

תבנית Spiral

כותרת

הנושאים בקורס מחולקים למקטעים של חומר המוצגים בסדר אשר אמור לקדם יכולות של פתרון בעיות אצל התלמידים. הרבה מקטעים של חומר מציגים נושא, אך אינם מכסים אותו לעומק. מידת הפירוט הניתנת תחילה מיועדת ליצור הבנה בסיסית שניתנת ליישום בפתרון בעיות. מעגלים נוספים מכילים מקטעי חומר תומכים שנכנסים לפירוט נוסף של החומר הנלמד.

בעיה/סוגייה

נושאים בקורס הינם רוב קשורים ביניהם בדרך זו או אחרת. לעתים קרובות יש להציג מספר נושאים שונים כדי להקנות לתלמידים מספיק כלים על מנת שיוכלו לעסוק בפתרון בעיות מעניינות. לעתים ניסיון להציג נושאים אלו בסדר "הגיוני" מוביל להתעסקות בהרבה פרטים וגורם לשעמום אצל התלמידים. חשוב לכוון את תהליך הלמידה כך שהתלמידים ירכשו כלים שיאפשרו להם לעסוק בפתרון בעיות משמעותיות כבר בשלבים התחלתיים של הקורס.

קהל/הקשר

לתבנית זו יש יישום בכל קורס בו יש מספר גדול של מושגים שיש לטפל בהם בו זמנית. התבנית מתאימה לשימוש בקורסי מבוא, בקורסי תכנות, קורסי ניתוח ופיתוח, ובקורסים מסוימים בטכנולוגית עצמים.

כוחות

נושאים גדולים כגון תכנות או תכנון דורשים שליטה בפרטים רבים. הוראה סדרתית עלולה להתבצע ללא יכולת להציג באופן שוטף בעיות מעניינות בגלל מחסור בכלים ושיטות. תלמידים אוהבים לבנות תוצרים

ולראות כיצד חלקים מורכבים ליצירת תוצר שלם. הם משתעממים בקלות כאשר המורה מתעכב יותר מדי על נושא אחד או על נושאים קרובים תלמידים יכולים להשתמש בכלים גם אם אין להם שליטה מלאה בהם. הם משתעממים גם בפתרון בעיות "מלאכותיות" שתכליתן להמחיש פרטים ומושגים שתכליתן לא נראית משמעותית.

פתרון

ארגן את הקורס כך שהצגת הנושאים תהיה הדרגתית במעגלים (ספירלית) מבלי לכסות את מכלול הפרטים בהצגה הראשונית ובהצגות הבאות פרט להצגה האחרונה. מומלץ להציג מספר נושאים כבר בהתחלה ולאתגר תלמידים בפתרון בעיות משמעותיות כבר תוך כדי הצגתם הראשונית. יש להקנות מספר רב של כלים מוקדם ככל האפשר ככל האפשר מבלי להעמיק בהם. המורה יחזור במהלך הקורס בכל מעגל של ההוראה הספירלית לטפל בנושאים ביתר הרחבה והעמקה ויטפל בהיבטים נוספים של הכלים שהוצגו קודם לכן. באופן זה שליטת התלמידים בכלים תעלה בהדרגה, והם יתנסו בפתרון בעיות מעניינות ומשמעותיות כבר בתחילת הקורס.

דיון/תוצאות/יישום

שימוש בתבנית מוביל להבנה משמעותית הנרכשת באופן הדרגתי של הנושאים הנלמדים וכן לפיתוח יכולות של פתרון בעיות. נושאים "גדולים" מטופלים תוך כדי לימוד נושאים "פשוטים" יותר הקשורים לנושא הגדול. לצורך ארגון ההוראה, המורה חייב לבחור חומר המכסה קבוצת נושאים הקשורים ביניהם, ולבחור מקרים פשוטים מכל נושא שיוצגו תחילה. תת קבוצה של כלים לפתרון בעיות המקיפה את הנושאים הנ"ל תוצג ותתורגל. בהמשך יוצג נושאים נוספים וכלים נוספים ובו זמנית יועמקו הנושאים שהוצגו קודם. גם אוסף הבעיות יורחב ומורכבותם תועלה בהדרגה. חשוב כל הזמן להדגיש את הקשר בין הנושאים הנלמדים.

התלמידים יקבלו בדרך זו את התחושה כיצד מרכיבים משולבים יחד כדי לקבל תוצר. תוצאה שלילית עבור תלמידים מסוימים עלולה להיות שהמורה יאלץ לדחות לשלב מאוחר יותר שאלות מסוימות המתייחסות לפירוט יתר של הנושאים המוצגים לא בשלמותם.

משאבים מיוחדים

על המורה להשקיע מחשבה רבה וזמן בארגון הלמידה. עליו לקבל החלטה לגבי סדר הצגת הנושאים, אילו פרטים יוצגו בהתחלה ואילו יידחו להמשך לטיפול במעגלים מאוחרים יותר של הצגת החומר. המורה צריך לברור תת נושאים שהכלים הקשורים אליהם מאפשרים באופן משולב לפתור בעיות משמעותיות. אפשר לנסח תחילה בעיות ולברור את הכלים הדרושים לפתרון ולתכנן כיצד להציג נושאים הקשורים לכלים הנדרשים. בהמשך אפשר להרחיב את הבעיות ולהרחיב את הכלים הדרושים לקבלת פתרונות כלליים יותר, וללמד נושאים נוספים הקשורים לכלים החדשים.

קשר עם תבניות דומות

Early Bird – בתבנית זו אפשר להיעזר להצגת המעגל הראשון של ההצגה הספירלית.
Test Tube – בתבנית זו ניתן להיעזר כדי להימנע מדשדוש בנושא ומכניסה לרמת פירוט מיותר במעגל נוכחי של ההצגה הספירלית.

Toy Box – בתבנית זו ניתן להיעזר כדי לבנות סדרת תרגילים ברמת קושי עולה.
Larger Than Life ו-*Lay of the Land* – יכולות לשמש כדי לתת מבט על כללי על הנושאים הנלמדים.

Fixer Upper – בתבנית זו יעשה שימוש כדי להציג את החומר החדש בכל מעגל.
אם הרעיון המרכזי הוא מורכב, ניתן להשתמש בתבנית *Toy Box* כדי להדגים את הרעיון.
רעיונות הקשורים זה לזה אפשר להציג תוך שימוש ב-*Tool Box*.

דוגמאות מוכרות ליישום מוצלח

בתבנית זו אפשר להשתמש בהצלחה להוראת תכנות ברמה בסיסית. למשל, במעגל ראשון אפשר להתחיל עם טיפוס *integer*, הוראות השמה, צורות פשוטות של מבנה בקרה *if* ו-*while* ולפתור בעיות פשוטות. במעגל השני אפשר להרחיב את הדיון בכל אחד מהנושאים- למשל, הצגה של *else*, לולאות אינסופיות, ועוד.

הסתייגויות

אין להשתמש בתבנית זו בקנה מידה "קטן". אם מחליטים ליישם אותה, יש לעשות זאת לאורך הקורס.

מקורות

הספרים הבאים נכתבו בגישה ספירלית:

Karel the Robot, Richards Pattis, Wiley, 1981.

ספר זה מציג את המעגל הראשון ללימוד פסקל.

Karel++, Joseph Bergin, Mark Stehlik, James Roberts, Richard Pattis, Wiley, 1997.

ספר זה מציג את המעגל הראשון ללימוד שפה מונחית עצמים.

Dana Anthony's patterns presented in PloP'95.

<http://st-www.cs.uiuc.edu/~chai/writing/classroom-ed.html>

תבנית Consistent Metaphor

כותרת

כאשר מלמדים נושא מורכב הנמצא מחוץ לתחומי ההתנסות אליהם רגילים התלמידים, מומלץ למצוא מטאפורה מורכבת וקונסיסטנטית עבור הנושא הנלמד, אשר בבסיסה מוכרת לתלמידים.

בעיה/סוגייה

במיוחד כאשר מלמדים תלמידים מתחילים, עלול להיווצר מצב של "ללכת לאיבוד" בים של פרטים של נושא נלמד מורכב. התוצאה – תלמידים עלולים שלא לראות כיצד נושא זה קשור לעקרונות ולמטרות מרכזיות יותר של התחום הנלמד. תלמידים מתחילים מתקשים לעשות סינטיזה של פרטים כדי לנתח ולנבא התנהגות של מערכת.

קהל/הקשר

כאשר מלמדים נושא מורכב הבנוי מהרבה פרקים שחלקם מאד מפורטים, נושא הנמצא מחוץ לעולם המושגים והניסיון של התלמידים, או שהוא ברמה טכנית גבוהה, כדוגמת חשיבה מונחית עצמים, יש לספק לתלמידים דרך להסתכל על הנושא כמכלול, מעבר לפרטים.

כוחות

המורה רוצה לתת לתלמידים "קיצור דרך" משמעותי קונסיסטנטי לחשיבת-על בהקשר לנושא מורכב. אותו "קיצור דרך" צריך לקשור את הנושא הנלמד לניסיונו של הלומד. תלמידים עלולים ללכת לאיבוד בים של פרטים וקשה להם לראות את הקשר בין הפרטים ואת "התמונה הגדולה", במיוחד כאשר הפרטים הם חדשים ובלתי מוכרים להם. עוזר מאד אם בלימוד נושאים חדשים רעיונות חדשים מתקשרים לרעיונות ידועים. לימוד כל פרטי הנושא עלול להיות תהליך ממושך. אחת ממטרות ההוראה היא להקנות לתלמידים יכולת הבנה של התנהגות מערכת וניבוי על אופן ותוצאות פעולתה.

פתרון

צור מטפורה קונסיסטנטית עם הנושא הנלמד במובן שהאינטראקציה בין מרכיביה דומה לזו בין מרכיבי הנושא. המטפורה תשמש את התלמידים לחשוב על הנושא באמצעותה. המטפורה צריכה לאפשר לתלמידים להסיק מסקנות נכונות לגבי החומר הנלמד תוך כדי שימוש בה.

דיון/תוצאות/יישום

המורה חייב לדעת את גבולות המטפורה ולהבהירם לתלמידים בכדי למנוע שימוש לא נכון בה, או הסקת מסקנות לא נכונות באמצעותה. ניתן להיעזר במטפורות להוראת אלמנטים קטנים של נושא, או כדי לתת מבט כללי על הנושא. בסיס המטפורה חייב להיות ידוע לסטודנטים.

משאבים מיוחדים

רצוי לקיים קבוצות דיון בין מורים לזיהוי מטפורות והערכת מגבלותיהן ומידת תרומתן.

קשר עם תבניות דומות

ניתן להשתמש בתבנית זו כדי ליישם את התבנית *Role Playing*.

דוגמאות מוכרות ליישום מוצלח

דרך מאד משמעותית ללמד תכנות מונחה עצמים היא לדמיין שכל עצם במערכת הוא אדם. למטפורה זו יש הרבה משותף עם תפיסה מונחית עצמים:

1. אנשים ועצמים הם שחקנים אוטונומיים.
2. אנשים ועצמים מגיבים להודעות.
3. בשני המקרים אופן הטיפול בהודעה תלוי במקבל ההודעה ולא בשולח.
4. יש טיפוסים שונים של אנשים (כמו של עצמים). הודעות יכולות להישלח בין אנשים/עצמים מטיפוסים שונים (לאו דווקא בין אנשים/עצמים מאותו טיפוס).

וכדומה.

מגבלות המטפורה: מערכת האנשים היא מקבילית בעוד שבמערכת לא-מקבילית האינטראקציה בין העצמים היא סדרתית.

דוגמא למטפורה מופשטת – מודלים פשוטים שנוהגים להשתמש בהם לתיאור המתרחש בזכרון המחשב בזמן תהליכי חישוב של שפות אימפרטיביות. מודלים אלו לרוב עושים את העבודה.

תבנית Toy Box

כותרת

מטרת התבנית לתת לתלמידים ידע היסטורי וטכנולוגי נרחב של התחום על ידי שנותנים להם "לשחק" עם כלי המחשה פדגוגיים.

בעיה/סוגייה

לעתים קרובות אין לתלמידים מושג ריאלי של מרחב אפשרויות היישום של מדעי המחשב או של הבסיס התיאורטי מדעי של התחום. היות ואנו מתמקדים לרוב בהרבה פרטים לעומק של הנושא הנלמד, קשה לעתים להקנות לתלמידים גם את מבט הרחב הרצוי.

קהל/הקשר

בתבנית זו יש אפשר להשתמש במספר קורסים ברמות שונות, בקורסי תכנות כמו גם בקורסים תיאורטיים.

כוחות

תלמידים צריכים להתמודד עם כמות גדולה של פרטים. הרבה יישומים של מדעי המחשב הם מאד מורכבים ונמצאים מחוץ לתחום המושגים של תלמידים. כאשר אנו מלמדים תכנות, כדאי לתת לתלמידים לתכנת עם כלים באמצעותם ניתן ללמד על רעיונות בהם הם יפגשו בהמשך לימודיהם בתחום. תלמידים חייבים לתכנת עם "משהו". לרוב בתחילת הלימוד הם עובדים עם מספרים שלמים וממשיים. ניתן ליצור עניין אצל תלמידים תוך שימוש באמצעים אחרים אטרקטיביים, ורצוי כאלו שבאמצעותם ניתן ללמד ברמות שונות באופן סימולטני. אנו רוצים לנצל את זמן התקשורת שלנו עם תלמידים באופן במידת יעילות מרבית. מצד שני, תלמידים צריכים לעבוד על בעיות.

פתרון

הכן שלדים של אפליקציות, בתחומים שונים. כל שלד ייצג באופן פשוט ומוחשי רעיון מרכזי שרוצים להעביר לתלמיד, ויהווה בסיס של מסגרת בה יפעל התלמיד ויתרגל. האפליקציות יבנו כך שיהיה ביניהן קשר שיוביל להבנת הקשרים בין המושגים המרכזיים. חבר משימות שימחישו מה ניתן לעשות תוך שימוש בעקרונות מדעי המחשב ומה חשיבות מדעי המחשב. תן לתלמידים ספריות של כלים מוכנים שניתן להשתמש בהם כדי לממש פעולות מורכבות. כלים אלו מהווים אבני בנין לפיתוח אפליקציות ומשמשים את התלמידים לבניית תוצרים. במקום עבודה עם מספרים שלמים או ממשים אפשר לפעול על עצמים יותר משמעותיים/מעניינים כגון שערים לוגיים. ניתן לאפשר לתלמידים להרחיב את הכלים המוגדרים מראש במידת הצורך כדי לפתח אפליקציות.

דיון/תוצאות/יישום

גישה זו מתאימה במיוחד לקורסים בתכנות מונחה עצמים. במקרה זה המורה יספק לתלמידים ספריות של מחלקות שימושיות אשר ישמשו את התלמידים לפתח תוצרים. ספריות אלו הן לרוב מורכבות בהרבה ממה שתלמידים יכולים לפתח באופן עצמאי והן מאפשרות לתלמידים להתנסות באופן פעיל בעבודה עם תכניות גדולות ומורכבות מעבר למה שיכלו לפתח בכוחות עצמם. בחירה מחושבת של הכלים על ידי המורה יכולה לתרום לתלמידים הבנה רחבה של התחום. כאשר ההיררכיה בין הכלים (המחלקות) בנויה היטב, היא יכולה לשמש מודל טוב לתלמידים לבנות מחלקות והירארכיות בהמשך, גם אם המטרה המיידית אינה ללמד תכנות מונחה עצמים.

משאבים מיוחדים

המורה צריך ליצור מבעוד מועד ספריות של מחלקות שימושיות אשר ישמשו את התלמידים לפתח תוצרים. מומלץ לחלוק חומרים עם מורים עמיתים.

קשר עם תבניות דומות

תבנית זו יכולה לספק דוגמאות למעגלים של *Spiral*.
אוסף כזה יכול להיות *Larger Than Life* במיוחד אם הקשר ביניהם חזק במיוחד.
אוסף שנבחר בקפידה יכול גם להדגים *Lay of the Land*.
Inlook-Outlook יכול לעזור לך להחליט אילו דוגמאות אתה צריך.

דוגמאות מוכרות ליישום מוצלח

אוסף מחלקות המממשות שערים לוגיים ומעגלים.
סימולטור מופשט של שפת אסמבלי.
סביבה להמחשת מקביליות.
משחק אינטראקטיבי מבוסס למידת מכונה.
אוסף מחלקות למימוש בסיס נתונים יחסי פשוט הניתן לתשואל על ידי שפה פשוטה.

הסתייגויות

אין להשתמש בתבנית זו מבלי להיכנס למחויבות של השקעת זמן גדולה ומקורות מתאימים. חיפוש ויצירת חומרים מתאימים דורשים זמן רב.

מקורות

The Analytical Engine : An Introduction to Computer Science Using the Internet,
Rick Decker, Stuart Hirshfield, PWS, 1998.

תבנית Tool Box

כותרת

הכוונה לתת לתלמידים לבנות במהלך קורס ערכת כלים על מנת שיוכלו להשתמש בהם בהמשך הקורס ובקורסים מתקדמים. אם התהליך מתוכנן ומתבצע כהלכה, הוא יכול להוות בסיס לרכישת הרגלים של שימוש חוזר בקוד. התלמידים עוברים תהליך של חונכות עד רכישת מיומנויות ברמה גבוהה.

בעיה/סוגייה

תלמידים לרוב פותרים כל בעיה במובן של פיתוח כל התכנית מן היסוד ועם סיום הפתרון מוחקים את התכנית ואינם חוזרים להשתמש בה יותר. כתוצאה מכך, הם אינם עושים שימוש חוזר בקוד, למרות שלעתים הם כותבים תכניות מאד דומות לאלו שכתבו בעבר.

קהל/הקשר

תבנית זו מתאימה בעיקר בקורסים לקורסי תכנות בסיסיים, ובעיקר לקורס מבני נתונים. עם זאת, ניתן להשתמש בה גם בקורסים מתקדמים יותר כמו בסיסי נתונים, בינה מלאכותית, מערכות הפעלה, קומפילרים, ועוד.

כוחות

תלמידים בקורסים מתקדמים יכולים לעשות שימוש בידע קודם הנרכש בקורסים קודמים. בפועל, הרבה תלמידים מעדיפים בנות מן היסוד מודולים שימושיים במקום לחפש מודולים קיימים זהים/דומים. כאשר לתלמיד יש ערכת כלים אישית לשימוש חוזר המלווה אותו, הוא ייעזר בה לפיתוח פרויקטים מורכבים וייעשה שימוש חוזר בקוד קיים.

בנייה של ערכה כזאת מפתחת מיומנויות והרגלי עבודה. למשל, ערכת כלים זקוקה לתחזוקה שוטפת. המרכיבים שלה צריכים להיבנות מתוך שאיפה לכלליות. התלמידים מתרגלים לעבוד תוך הסתרת מידע.

פתרון

יש לחבר משימות לתלמיד שמכילות סעיפים. בכל משימה יש סעיף המתייחס לבניית כלי כללי שאפשר להשתמש בו גם בפרוייקטים אחרים, ולכן יש להנחת אל התלמידים לפתח אותו כך שיהיה נוח לעשות בו שימוש חוזר. התכנון לשימוש חוזר צריך להיות מפורש ויש לערוך דיונים בכיתה על כך. יש לעודד פיתוח כלים בצוות, ואינטגרציה ושיפור של כלים אישיים שפותחו באופן פרטני. בניית יישומים תוך שימוש בכלים שפותחו על ידי תלמידים אחרים.

דיון/תוצאות/יישום

תלמידים רוכשים מיומנויות כבר בקורסי מבוא על ידי בניית מודולים שניתן לעשות בהם שימוש חוזר. יש ללמד שפות התומכות בשימוש חוזר בקוד, ורצוי בהכללה (templates). מורים יכולים לספק חלק מהכלים כבנויים מראש. אין צורך שהתלמידים יפתחו הכל לבדם. הם יכולים להשלים את מה שהמורה נתן כבסיס. יש להקדיש זמן להערכת הכלים מבחינת נכונות, והפוטנציאל שלהם לשימוש חוזר. מורים המלמדים בקורסים מתקדמים יותר צריכם להיות מודעים לכך שלתלמידים יש את ערכות הכלים, ועליהם לחבר משימות שמהן יתבקש לעשות שימוש בערכות הכלים כמו גם להרחיב ערכות אלו. בהתאם, תיאום בין מורים חשוב במיוחד.

משאבים מיוחדים

הקדש מחשבה רבה לתכנון קורסי המבוא, כך שהכלים שהתלמידים יפתחו במהלכם יהיו בעלי פוטנציאל שימוש גבוה גם בהמשך. אחד השיקולים צריך להיות אי-תלות בפלטפורמה. עליך להכין כלים שיינתנו לתלמידים כבסיס לכלים שהם ייצרו בעצמם. ערכת הכלים המתפתחת שיבנה התלמיד תכיל גם את הכלים המוכנים מראש שיינתנו לו על ידי המורה וגם את הכלים שהתלמיד יפתח.

קשר עם תבניות דומות

המורה יכול להשתמש ב- *Fixer Upper* על מנת לספק כלים עם ליקויים קלים אותם יתבקשו התלמידים לתקן. פיתוח הכלים ועידונם יכול להיעשות במסגרת כל מעגל של *Spiral*.

דוגמאות מוכרות ליישום מוצלח

דוגמא טובה לשימוש בתבנית זו מופיעה ב- *Kernighan & Plauger's Software Tools*. מוצג שם כיצד ניתן להשיג דרגה גבוהה של שימוש חוזר באמצעות כלים מאד פשוטים. אפשר להשתמש בתבנית בטיפול במבני נתונים שעושים להם שינוי קל. שימוש בשפות שמבוססות על עבודה בספריות.

הסתייגויות

תבנית זו אינה מתאימה כפי הנראה להוראת קורסים תיאורטיים.

מקורות

Software Tools in Pascal, Brian Kernighan, P.J. Plauger, Addison Wesley, 1981.

בניית מערך שיעור ופעילויות ליישום תבנית Tool Box

נטלי ויסברג ושרית חיים

בחרנו בתבנית הפדגוגית – "Tool Box" – פרט להעברת ידע, התלמיד יהיה מסוגל לפתח כלים בהם יוכל להשתמש בשלב מאוחר יותר.

מערך שיעור

קהל היעד: תלמידי מדעי המחשב ברמה של יסודות 1 – תלמידי כיתה י.

משך השיעור: 45 דקות.

הנושא הנלמד: חזקות

רקע קודם: קלט, פלט, לולאת For, חזקה ריבועית.

שיטת ההוראה: בשיעור פרונטלי בכיתה תוצג הבעיה, יינתן תרגיל דוגמא, יוצע הפתרון לפונקציה, ויוצע פתרון לתרגיל הדוגמא.

על-פי התבנית הפדגוגית בשיעורים פרונטליים בכיתה ילמדו פונקציות ספציפיות (לדוגמא – חזקה), שאינן מובנות בשפה, בצורה מעמיקה הכוללת הסברים ודוגמאות. כל פונקציה, לאחר שנלמדה, תשמר ביחידת ספרייה.

לאחר שהפונקציות יובנו וייושמו על-ידי התלמיד במעבדה, תשמש אותו הפונקציה ככלי. כלומר, בעת הצורך, בכתיבת תכנית, ישתמש התלמיד בפונקציה שלמד (ישלוף את הפונקציה מתוך ארגז הכלים – יחידת הספרייה) מבלי לכתוב את תכנה מחדש.

הצגת הבעיה – 5 דקות

במטרה לפתור בעיות מתמטיות שונות אנו נתקלים בצורך בשימוש בהעלאה בחזקה. הפקודה חזקה ריבועית מובנית בשפה (התלמידים אינם נדרשים לכתבה) ומוכרת לתלמידים, גם משימוש בה.

סיעור מוחות – 15 דקות

דוגמא לתרגיל הדורש שימוש בחזקות:

פתח וישם אלגוריתם אשר הקלט הוא מספר N שלם והפלט הוא סכום הטור הבא:

$$S=2^0+2^1+2^2+\dots+2^{N-1}+2^N$$

דיון, הסקת מסקנות ורישום הפונקציה – 15 דקות

פונקציה לחישוב חזקה אשר לה שני פרמטרים, האחד בסיס החזקה והאחר מעריך החזקה.

הפרמטרים: basis – ישמור את ערכו של בסיס החזקה

exponent – ישמור את ערכו של מעריך החזקה

הפרמטרים הפנימיים: P – ישמור מכפלה מצטברת

Function power (basis: integer; exponent: integer): integer;

Var

P: integer;

I: integer;

Begin

P:=1;

For I:=1 to exp do

P:=p*basis;

Power:=p;

End;

פתרון הדוגמא מתחילת השיעור – 15 דקות

אלגוריתם מילולי:

1. אתחל temp ב-1 $\{1=2^0\}$

2. עבור I מ-1 עד n בצע

Temp←temp+power (2,I) 2.1

התכנית:

Function S (n: integer): integer;

Var

I: integer;

Temp: integer;

Begin

Temp:=1;

For I:=1 to n do

Temp:=temp+power (2,I);

S:=temp;

End;

שעורי בית:

1. כתוב אלגוריתם מילולי לפונקציה לחישוב חזקה אשר הוצגה במהלך השיעור.

2. פתח וישם אלגוריתם אשר הקלט הוא מספר N שלם כמעריך החזקה ובסיס החזקה הוא

מספר זוגי החל מהמספר 2. הפלט הוא סכום הטור הבא:

$$S=2^0+4^1+6^2+\dots+x^{N-1}+(x+2)^N$$

תבנית Lay of the Land

כותרת

תלמידים מתנסים בבחינת תוצר גדול שהוא מעבר ליכולת הפיתוח שלהם במטרה להראות להם את מורכבות התחום אותו הם אמורים ללמוד.

בעיה/סוגייה

לעתים קרובות אנו מלמדים נושאים שמכסים כמות רבה של חומר. אם הסטודנטים לא רואים את "התמונה הגדולה" מספיק מוקדם, יתכן והם לעולם לא יראו אותה מפאת ניסיון להבין ים של פרטים. מטרתנו להראות לתלמידים ראייה רוחבית של נושא גדול על מנת שיהיו להם נקודות ייחוס שימנעו מהם ללכת לאיבוד במהלך התוודעות לפרטים הרבים לאורך התקדמות הקורס.

קהל/הקשר

לתבנית זו יש יישום בכל קורס בו יש מספר גדול של מושגים ותת נושאים, במיוחד כאשר חשוב לקשור ביניהם באופן מסוים. למשל, לימוד תכנות, לימוד תכנון.

כוחות

הוראה היא לעתים קרובות הדרגתית, תוך הצגת נושאים זה אחר זה. זו אינה הדרך הטובה ביותר ללמוד במיוחד כאשר המורה מנסה לאכוף סדר נוקשה של הצגת מושגים, והיא נוגדת את הגישה הקונסטרוקטיביסטית.

תלמידים צריכים לראות את התמונה הגדולה מלבד לפרטים.

בשלב התחלתי תלמידים יכולים לייצר תוצרים פשוטים בלבד, אולם, הם בהחלט יכולים לבחון (גם אם באופן מלאכותי) תוצר מורכב. רוב האנשים יכולים לקרוא ולהבין משהו שהוא מורכב יותר מאשר שהם יכולים לייצר בכוחות עצמם.

ראיית התמונה הכללית יכולה לתת לתלמידים מוטיבציה ללמוד חלקים שונים של החומר, מכיוון שיש להם מושג כללי איך ניתן להשתמש בהם. זה נכון במיוחד אם המורה מציג את התמונה הכללית תוך שימוש בדוגמה מעניינת.

פתרון

תן לתלמידים תוצר גדול לבחינה בשלב מוקדם של הקורס. הם יראו בדרך זו את התכלית של לימוד הקורס, ומהי נקודת המטרה אליה הם אמורים לשאוף. התוצר צריך להיות ברמה מתאימה אליה התלמידים אמורים להגיע בסיום הקורס, דהיינו בקנה מידה של מורכבות שעליהם להיות מסוגלים לבצע בסופו של דבר. התוצר צריך להכיל מכלול מרכיבים המייצגים את הנושאים השונים הנלמדים בקורס. רצוי שהתוצר יכיל נקודות רגישות (בעייתיות) בהן יתעורר הצורך לדון.

מעבר להצגת התוצר בתחילת הקורס ובהינתן הראשונית, המורה ישתמש בתוצר כדוגמה מלווה לאורך כל הקורס וידון אם תלמידיו על היבטים שונים של התוצר בהקשר לנושאים הנלמדים בקורס, ועל מגבלותיו.

דיון/תוצאות/יישום

שימוש בתבנית מוביל לכך שתלמידים יראו את תכלית לימוד הקורס. יהיה להם מודל עליו הם יוכלו לבסס את עבודתם.

יש להכין את התוצר מבעוד מועד. אפשר להיעזר בפרויקטי סיום של תלמידים משנים קודמות, ולשנותם כל שיתאימו כדוגמה מלווה, תוך הדגשת הנקודות החשובות לדיון.

משאבים מיוחדים

פרויקטים לדוגמא, בעלי מבנה וסגנון המוערכים כמעולים.

קשר עם תבניות דומות

Larger than Life – בתבנית זו גם עוסקת בהצגת תוצר מורכב בשלב מוקדם של הלימוד, אך לא רק לצורך בחינה ודיון, כי אם לצורך שינוי.

Spiral – בתבנית זו ניתן להיעזר כדי לבחון את המוצר בהדרגה במעגלים עוקבים.

התוצר יכול להיות *Fixer Upper*.

אם התוצר נבחר בקפידה הוא יכול לשמש גם כ- *Early Bird*.

Spiral – בתבנית זו ניתן להיעזר כדי לבחון את המוצר בהדרגה במעגלים עוקבים.

התוצר יכול להיות *Fixer Upper*.

ניתן להשתמש ב- *Test Tube* כדי לאפשר לתלמידים לתת מענה עצמי לשאלותיהם אודות התוצר.

דוגמאות מוכרות ליישום מוצלח

תכנית גדולה בשפה תומכת תכנות מונחה עצמים הכוללת מספר מחלקות הקשורות ביניהן באופן מעניין יכולה לשמש כתוצר דוגמה בקורס תכנות מבוסס גישה מונחית עצמים. מסמך תכנון מפורט שלם יכול לשמש כתוצר לדוגמא בקורס תכנון. דיאגרמת E-R מורכבת וטבלאות תואמות המתארים בסיס נתונים מורכב יכולים לשמש כתוצר לדוגמא בקורס בסיסי נתונים.

תבנית Fixer Upper

כותרת

כאשר נותנים לתלמידים להתנסות בבחינת תוצר גדול שבעקרון בנוי כהלכה, אך שתולות בו בכוונה תחילה משיקולים דידקטיים מגרעות, משיגים שתי מטרות: הצגת נושא מורכב בשלב מוקדם, והתנסות מוקדמת בניתוח שגיאות ותיקון.

בעיה/סוגייה

לעתים קרובות תלמידים עובדים על "בעיות צעצוע" קטנות מכיוון שאין להם ניסיון ויכולת לבנות תוצרים גדולים מהיסוד. בעיות אמיתיות הן "בעיות גדולות" והן המשמעותיות עבור התלמיד. אנו מעוניינים לאפשר לתלמידים לעבוד על תוצרים גדולים מבלי להמם אותם. מאידך, תלמידים מתקשים כאשר שגיאות בלתי צפויות צצות במהלך עבודתם. הם מתקשים להבין ולטפל בהודעות שגיאה.

קהל/הקשר

לתבנית זו יש יישום במספר קורסים וברמות שונות. ניתן להשתמש בה בשלבים מאד התחלתיים של קורסי תכנות ובלמוד של ניתוח ותכנון, ופתרון בעיות.

כוחות

לעתים קרובות אנו צריכים להציג לתלמידים תחום חדש הדורש מיומנויות במספר נושאים. תלמידים לרוב לא מצליחים לראות כיצד מכלול הנושאים מתקשרים זה לזה עקב הצגה סדרתית שלהם. הם גם לרוב אינם מצליחים להבין את המשמעות של איתור ותיקון שגיאות. תיקון של תוצר גדול על ידי תלמידים נותן להם הבנה יותר טובה תוך התמקדות בבעיות מעניינות, ומאפשר להם לבצע אינטגרציה של מספר נושאים לצורך פתרון של בעיה אחת. תלמידים יכולים להיתרם ממפגש עם בעיות גדולות שרמתן מעבר ליכולת לפתור אותם באופן עצמאי מן היסוד בשלב הנוכחי של לימודיהם. הם צריכים לפתח יכולות ניתוח והערכה של תכניות, תכנונים וכדו'.

פתרון

תן לתלמידים תוצר גדול לבחינה בשלב מוקדם של הקורס. התוצר אמור להוות פתרון נכון במרביתו לבעיה מוגדרת, אך מכיל שגיאות ו/או ליקויים שהוכנסו במתכוון על ידי המורה מתוך שיקולים דידקטיים. התוצר צריך להיות מספיק גדול ולהכיל מספר ליקויים, רובם פשוטים וניתנים לזיהוי בקלות יחסית, וחלקם יותר מורכבים. התלמידים מתבקשים לזהות ולתקן את הליקויים ולדון במאפייניהם ובאופן תיקונם. לבסוף, יש לבקש את התלמידים לדון במבנה הכולל של התוצר, ולהסיק מסקנות.

דיון/תוצאות/יישום

התבנית מאפשרת לתלמידים לעבוד באופן פעיל על תוצרים גדולים שהינם מעבר ליכולת הפיתוח העצמי המלא שלהם בשלב זה של לימודיהם. איתור שגיאות וליקויים וטיפול בהם בתוצר שלא הם פיתחו, עשוי לתרום לבדיקת נכונות ואיכות של תוצרים שהם מפתחים.

חשוב מאד שהתוצר בכללותו יהיה בנוי כהלכה, ושהליקויים בו יהיו נקודתיים. דרך אפשרית לבנות כזה תוצר הוא להתחיל מתוצר טוב ללא ליקויים, ולהכניס בו ליקויים בהתאם לשיקולים דידקטיים. מהות הליקויים תלויה בשלב בו התוצר מוצג לתלמידים. בשלבים מוקדמים של הלימוד הליקויים יהיו מינוריים וייתייחסו לפרטים קטנים. בשלב מאוחר יותר של הלימוד אפשר להציג גם ליקויים במבנה של התוצר. ניתן להשתמש בזיהוי הליקויים כמנוף לניהול דיונים פוריים וגילויים חדשים.

משאבים מיוחדים

בעיה מעניינת ופתרון בנוי כהלכה אותו יש "לקלקל" במידה ולשתול את השגיאות בהן רוצים לדון. ניתן להשתמש בפרויקטים מעולים משנים קודמות, ספרים מקצועיים, וכדו'.

קשר עם תבניות דומות

כאשר הפרויקט מוצג בתחילת הקורס והוא מייצג את הרעיונות המרכזיים של הקורס, הרי התבנית היא גם *Early Bird*.

כאשר הפרויקט הוא גדול מאד אז התבנית היא גם מקרה פרטי של *Larger than Life*.

כאשר הפרויקט מייצג נושא חשוב מסוים אז התבנית היא גם מקרה פרטי של *Toy Box*.

כאשר הפרויקט מייצג את הנושא הכללי של הקורס אז התבנית מדגימה את התבנית *Lay of the Land*. כאשר בתוצר המתוקן נעשה שימוש בהקשר רחב יותר התבנית יכולה להיות מקרה פרטי של *Tool Box*. חלק ממטרות התבנית משותפות לתבנית *Mistake* למרות ההבדל המשמעותי ביניהן: כאן התלמידים מתבקשים למצוא ולתקן שגיאות בתוצר קיים בעוד שב- *Mistake* הם מתבקשים ליצור במכוון שגיאות ספציפיות בתוצר שהם מפתחים.

תבנית זו היא דרך להשיג *Read before Write*.

דוגמאות מוכרות ליישום מוצלח

בתבנית זו נעשה שימוש להוראת: תכנות למתחילים, מבוא לתכנון, פיתוח פרויקטים בצוות.

הסתייגויות

תלמידים עלולים להיות מתוסכלים מטיפול בתוצרים "גדולים מדי". המורה צריך להיות מוכן לתמוך ולעודד את התלמידים ולשדר להם שכך נראות בעיות אמיתיות ותוצרים בעולם האמיתי. הוא צריך לשכנע את התלמידים שזה בסדר להיות בהתחלה מתוסכל וללא ידיעת כל הכלים הדרושים לטיפול בתוצר כה גדול.

תבנית Larger than Life

כותרת

תלמידים הלומדים תכנות מונחה עצמים ותכנון יכולים וחשוב שיבחנו מערכות תוכנה מוכנות הרבה לפני שהם יכולים לתכנן וליישם מערכות בקנה מידה כזה באופן עצמאי.

בעיה/סוגייה

תוצרים משמעותיים מבוססי פרדיגמת תכנות מונחה עצמים הם גדולים ומורכבים למדי. לרוב תוצרים אלו ב"עולם האמיתי" מפותחים על ידי צוותים ולא בעבודה עצמית של אדם אחד. לעומת זאת, מורים רבים סבורים שלתלמידים צריכים לפתח תוצרים רק באופן עצמאי, ואינם מעודדים עבודת פיתוח בצוות. כמו כן, רוב העבודה ב"עולם האמיתי" אינה כרוכה בפיתוח מן היסוד, כי אם בשינוי של תוצרים קיימים, או בהרחבה הדרגתית של תוצר קיים.

קהל/הקשר

בתבנית זו יש אפשר להשתמש בקורסים לתלמידים ברמות שונות, החל מתלמידים מתחילים וכלה בתלמידים מנוסים מאד. זוהי דוגמא של "קריאה לפני כתיבה".

כוחות

תלמידים אינם יכולים בתחילת הקורס לפתח תוצרים מורכבים, אך כדאי שיוצגו בפניהם תוצרים כאלו. הם יכולים לבחון תוצרים ברמה שהרבה מעבר ליכולת הפיתוח שלהם, וללמוד מכך הרבה. בעוד שזמן רב מוקדש בכיתה על פרטים ברמות הפשטה נמוכות ולפרטי פרטים, רצוי שהתלמידים יוכלו לראות את "התמונה הגדולה".

אם התלמידים בונים לבד כל כלי בו הם משתמשים, הם עלולים לקבל מבט מוטעה על כיצד מפתחים תוכנה ב"עולם האמיתי".

פתרון

תן לתלמידים גישה לתכניות גדולות ולמסמכי תכנון ברמה מעבר ליכולת הפיתוח העצמי שלהם. השתמש בתוצרים אלו כבסיס ליצירת תרגילים הכרוכים בעריכת שינויים קטנים והרחבות קטנות בתוצר הנתון, וכן בתיקון ליקויים מינוריים.

יש להקפיד שהתוצרים יהיו באיכות גבוהה שכן התלמידים יתייחסו אליהם כמודל לפיתוח עתידי. כמובן שאם המטרה להדגים תוצר שפיתוחו לקוי, יש להתייחס אליו בתור אי-דוגמא.

דיון/תוצאות/יישום

תלמידים צריכים להסתגל להשתמש בספריות שנכתבו על ידי אחרים. ספריות אלו הן לרוב מורכבות בהרבה ממה שתלמידים יכולים לפתח באופן עצמאי. הם עשויות גם להכיל קוד ברמה מושגית נמוכה שחבל להקדיש זמן לעסוק בו במהלך הקורס, ועדיף לקבל אותו כמוכן מראש. יחד עם זאת היבטים של תכנון קוד כזה יכולים להיות מעניינים לדיון.

משאבים מיוחדים

תוצרים גדולים באיכות גבוהה של תכנון ויישום.

קשר עם תבניות דומות

אם התוצר דורש תיקון, אז התבנית היא גם *Fixer Upper*.
אם התוצר מדגיש רעיונות מרכזיים והוא מוצג מוקדם, אז התבנית היא גם *Early Bird*.
אם התוצר שימושי מאד, אז התבנית מובילה ל- *Tool Box*.
אם התוצר מכסה את רוב המושגים המרכזיים בקורס, אז התבנית היא גם מקרה פרטי של *Lay of the Land*.

דוגמאות מוכרות ליישום מוצלח

הספרייה הסטנדרטית של C++ היא דוגמא לתוצר כזה. בדומה, ה-AWT של JAVA. מסמכי תכנון של מערכות גדולות הם דוגמא נוספת. בקורס קומפיילרים אפשר לתת קומפיילר מסוים כתוצר כזה, ובסוף הקורס לבקש לפתח קומפיילר עבור שפה אחרת.
בכל קורס מבוסס תכנות מונחה עצמים רצוי להשתמש בספריות בין אם הן סטנדרטיות, או נכתבו על ידי המורה, או על ידי אחרים.
Linda Rising ו-Owen Astrachan דיווחו על ניסיון הוראה מוצלח בשיטה זו.

מקורות

Rising, L. Removing the emphasis on coding in a course on software engineering, SIGCSE Bulletin, February 1989.

Astrachan, O., Berry, G., Cox, L. and Mitchener, G.. *Design Patterns: An Essential Component of CS Curricula*. SIGCSE Technical Symposium on Computer Science Education. Atlanta, GA, February 1998.

תבניות פדגוגיות שהוצעו על-ידי מורים למדעי המחשב

סעיף זה מכיל שתי הצעות של תבניות חדשות להוראת מדעי המחשב שהוצעו על ידי מורים למדעי המחשב, בצד הצעות לשימוש בהן ובתבניות שפיתח Joseph Bergin לתכנון פעילויות למידה.

נטלי ויסברג ושרית חיים:

הצעה של תבנית פדגוגית חדשה להוראת מדעי המחשב: "זה הקטן גדול יהיה..."
בניית מערך שיעור ופעילויות ליישום תבנית פדגוגית להוראת מדעי המחשב על-פי התבנית הפדגוגית –
"Tool Box"

מיכל בראל וזהבה לוי:

הצעה של תבנית פדגוגית חדשה להוראת מדעי המחשב: "זרימת מידע"
מערך שיעור ופעילויות ליישום התבנית הפדגוגית החדשה ותבנית פדגוגית על פי Joseph Bergin.

הצעה של תבנית פדגוגית חדשה להוראת מדעי המחשב

"זה הקטן גדול יהיה..."

נטלי ויסברג ושרית חיים

כותרת

ארגן את הלמידה סביב מטרות מרכזיות (פרויקטים) המניעות תלמידים למצוא תשובות. תהליך ההערכה של התלמידים יתבסס על בדיקת יכולתם של התלמידים ליישם ידע שלמדו לפתרון בעיה נתונה (פרויקט) ולהשגת מטרת העל.

קהל/הקשר

לתבנית פדגוגית זו שימוש נרחב במגוון מקצועות. התבנית מאפשרת להגיע לשלבים של יישום, ניתוח וסינטיזה בשלבים מוקדמים יחסית של תהליך הלמידה ברמות שונות ובהיקפי חומר משתנים.

כוחות

התלמידים יבינו כי מתוך הצורך מתרחשת למידה ותוך כדי יישום מתרחשת הפנמה של החומר הנלמד. תלמידים בדרך כלל זוכרים טוב יותר חוויות התנסותיות מאשר למידה בשיטה הפרונטלית, כשהם נדרשים להיות פאסיביים.

הבעיה/הנושא

- כדי לעודד תלמידים ללמוד, הדגש בלמידה אינו מושם רק בשינון החומר אלא גם בדרך של התנסות וביישום הידע.
- פיאז'ה, אצל שובל וגם דיואי הדגישו את חשיבות ההתנסות הפעילה כאמצעי לפיתוח חשיבה.
- תיאורית הלמידה של ויגוצקי הדגישה את האינטרקציה החברתית כגורם מכריע בתהליך הלמידה.
- סביבת לימודים המדגישה למידה קבוצתית, שיתוף פעולה בצוותים מסייעת ללמידה בניגוד לסביבת לימודים המשאירה את הלומד פאסיבי ונתון להאזנה תמידית למקור הידע הבלעדי – המורה.

ארגן את הקורס כך שתהיה ברשותך רשימת פרויקטים, כשכל פרויקט הוא סיכום של מספר נושאים הנלמדים ברצף, שיש ביניהם קשר. לאחר ביצוע כל הפרוייקטים התלמידים יקבלו פרויקט מסכם אותו יבצעו תוך שימוש בכל הכלים שנרכשו במהלך ביצוע הפרוייקטים לאורך הקורס.

פתרון

הצג בפני התלמידים את הפרוייקט הראשון (הבסיסי ביותר) ובקש מהם לחשוב מהם השלבים לבניית הפרוייקט ומהם הכלים להם הם זקוקים. כמו כן שיתנו דעתם לבעיות המתעוררות במהלך תהליך הבנייה והחשיבה.

יש לצפות לדרישה מצד התלמידים להקניית הרגלים וללימוד כלים אופרטיביים לביצוע הפרוייקט. על המורה לחשוב על מספר פרוייקטים, כשכל אחד מהם מהווה סיכום של מספר נושאים ומכיל שימוש במספר כלים, הקשורים זה בזה ושומרים להילמד בדרך כלל ברצף. כמו כן עליו לחשוב על פרוייקטים רחבים יותר המכילים בתוכם נושאים וכלים בהם נעשה שימוש בפרוייקטים במהלך השנה. הרעיון הוא שכשיגיע תלמיד לפרוייקט המסכם הוא יוכל לבחור לעצמו פרוייקט מתוך מגוון פרוייקטים שהמורה יציע לו (או לחילופין התלמיד יציע בעצמו) ובאמצעותו יבחן את הידע שרכש במהלך לימודיו לאורך שנת הלימודים.

את הפרוייקטים לאורך השנה מבצעים התלמידים יחד כצוות. את הפרוייקט המסכם ניתן לבצע עצמאית או בזוגות בלבד.

דיון/תוצאות/יישום

התלמידים רוכשים את הידע הבסיסי, כלים ומרכיבים שונים הניתנים לשימוש חוזר בשיעורים הראשונים, שהם שיעורים פרונטליים.

כל שיעור מחזק את הכלים שנרכשו בשיעור קודם ומציג כלים נוספים, חדשים, עליהם מושם הדגש באותו שיעור.

במסגרת העבודה על כל אחד מהפרוייקטים, התלמיד מתבסס על חלקים מפרוייקטים קודמים, המהווים עתה כלי בידו של התלמיד. כל כלי כזה הוא יחידה סגורה העומדת בפני עצמה והמאפשרת לתלמיד שימוש חוזר בה על מנת לפתח פרוייקטים נרחבים יותר שיהוו כלי חדש. הקשר בין הפרוייקטים לשיעורים הוא מרכיב חשוב בתבנית זו.

משאבים מיוחדים

על המורה לתת את הדעת לגבי תכנון השיעורים הראשונים תוך שימת לב מיוחדת להצגת הכלים הבסיסיים אבל בעיקר מהם הכלים וההרגלים להם יידרשו התלמידים בעתיד לבוא, בעת עבודתם על הפרוייקטים.

קיימת אפשרות, על-פי שיקול דעת המורה ובהתאם לרמת התלמידים בכיתה, לספק לתלמידים יישומים של כלים שלא פותחו בכיתה או/ו שאין עדיין באפשרותו של התלמיד את היכולת לבנותם בכוחות עצמו.

קשר עם תבניות דומות

Toy box – תבנית זו מספקת רצף תרגילים ברמות קושי ההולכות וגדלות.

Tool box – בתבנית זו יעשה שימוש במקרים בהם התלמיד יתבקש להשתמש/לבנות באופן עצמאי דבר מה ויזדקק לכלים אותם למד במסגרת שיעוריו בכיתה.

Spiral – בתבנית זו יעשה שימוש יעיל בנושאים מתפתחים. ככל שהנושא מתפתח יותר היקף החומר וההתעמקות בו ילכו ויגדלו.

בניית מערך שיעור ופעילויות ליישום התבנית הפדגוגית החדשה שהוגדרה

נתבסס על התבנית הפדגוגית – "זה הקטן גדול יהיה" – ארגון הלמידה סביב מטרות מרכזיות (פרויקטים) המניעות תלמידים למצוא תשובות.

בחרנו להציג את תהליך התפתחות התבנית הפדגוגית באמצעות דף ובו רשימת פרויקטונים. כל שאלה בדף מהווה מעיין פרויקטון העומד בפני עצמו אך גם מתבסס על הפרוייקטון שלפניו, על השאלה הקודמת. בכל שאלה מתבססים על ידע קיים ורוכשים ידע חדש תוך שימוש בפקודות חדשות.

דף העבודה אינו ניתן לתלמיד אך משמש את המורה בעבודתו ומלווה את תהליך ההוראה למשך מספר שיעורים, חלקם שיעורים פרונטליים וחלקם שיעורי מעבדה. בשיעורים הפרונטליים ירכוש התלמיד את הכלים של השפה (פקודות, מבנה וכו').

קהל היעד: תלמידי מדעי המחשב ברמה של יסודות 1 – תלמידי כיתה י.

רשימת פרויקטונים

פרוייקטון מס' 1 - חיבור שני מספרים שלמים.

ידע קודם – הגדרת משתנה מטיפוס שלם, קלט ופלט.

ידע נרכש – שילוב שלושת המרכיבים.

פרוייקטון מס' 2 - חיבור או חיסור של שני מספרים שלמים.

ידע קודם – הגדרת משתנה מטיפוס שלם, קלט ופלט ושילובם.

ידע נרכש – תנאי ומשתנה תווי.

פרוייקטון מס' 3 - חיבור או חיסור של יותר משני מספרים שלמים.

ידע קודם – הגדרת משתנה מטיפוס שלם, קלט ופלט ושילובם, תנאי ומשתנה תווי.

ידע נרכש – לולאת while.

פרוייקטון מס' 4 – פתרון מספר תרגילים משולבים הכוללים פעולות חיבור וחסור ברצף.

ידע קודם – הגדרת משתנה מטיפוס שלם, קלט ופלט ושילובם, תנאי ומשתנה תווי, לולאת

while.

ידע נרכש – קינון לולאות. אם ידוע מספר התרגילים מראש, לולאת for. אחרת לולאת

while.

פרוייקטון מס' 5 – פתרון מספר תרגילים משולבים הכוללים פעולות חיבור, חיסור, כפל וחילוק ברצף.

ידע קודם – הגדרת משתנה מטיפוס שלם, קלט ופלט ושילובם, תנאי ומשתנה תווי, לולאת

while ו/או for, קינון לולאות.

ידע נרכש – case ושגרות.

הצעה של תבנית פדגוגית חדשה להוראת מדעי המחשב

"זרימת מידע"

מיכל בראל וזהבה לוי

תלמידים לומדים שהדרך הנכונה לפתור בעיות מורכבות היא ע"י חלוקתן לתת משימות קטנות, בלתי תלויות זו בזו, ככל האפשר. כל תת משימה מטפלת בקטע אחר של הבעיה ויישומה דורש מספר מצומצם של משפטים בשפת התכנות. בתהליך זה תוצאות של תת משימה אחת הופכות לעיתים לקלט של תת משימה אחרת. לפיכך, יש לדאוג לזרימת מידע נכונה בין תתי המשימות השונות. זרימת המידע מתבצעת באמצעות מנגנון "העברת פרמטרים", תהליך המאפשר העברת מידע בין החלקים השונים של התוכנית.

הבעיה

הוראת השיטה של פירוק הבעיה לתת משימות ולימוד תהליך זרימת המידע בין החלקים השונים, חיוניים לפיתוח תוכניות מורכבות. יש ללמד את התלמידים כיצד לחלק בעיה לתת משימות ולהנחותם בזיהוי תהליכי זרימת המידע הנוצרים עקב הפירוק.

תלמידים מתחילים מעדיפים לכתוב קוד קצר לפתרון בעיה ברמת הפשטה אחת בלבד, ואינם נוטים באופן טבעי לחלק את הבעיה לתת המשימות. בנוסף, הם מתקשים בהבנת השוני בין הסוגים השונים של הפרמטרים (פרמטר של ערך / פרמטר של משתנה). עובדה זו עלולה ליצור קושי עתידי בהתמודדותם עם תוכניות הדורשות הבנה מעמיקה יותר של מנגנון זרימת המידע במחשב.

קהל היעד

תבנית זו ישימה לכל קורס תכנות, בו התלמידים מתבקשים לפתור בעיות באמצעות כתיבה מבנית (מודולארית) כלומר, תכנית המורכבת מתת תוכניות שונות המעבירות מידע ביניהן.

התבנית ישימה גם בשפות דקלרטיביות כמו בפרולוג, שכן, המתוארים הם כמו פרמטרים. כלומר, ערך של מתואר במתאר מסוים יכול לשמש כנתון לקריאה של מתאר אחר.

כבר בשלבים הראשונים, גם כאשר מדובר באלגוריתמים פשוטים, יש להקפיד על פיתרון בעיות ע"י פירוקן לתת משימות לפני היישום בשפת התכנות, זאת במטרה להטמיע את רעיון החלוקה. אמנם, בשלב זה עדיין אין משמעות לתהליך של זרימת המידע בין תת המשימות, אולם לא כך הוא בהמשך, כאשר האלגוריתמים הופכים למורכבים יותר.

הכוונה היא שהתלמיד יסגל לעצמו הרגלים נאותים, אותם יישם בשלב מאוחר יותר של פיתרון בעיות מורכבות במדעי המחשב.

כוחות

תלמידים אינם מכירים בחשיבות החלוקה לתת משימות ומעדיפים לפתור בעיות על ידי כתיבת הקוד בשפת תכנות. הם פועלים ברמת הפשטה אחת. אולם, כאשר הם נתקלים בבעיה מורכבת, הם מתקשים להתמודד איתה.

כאשר מאלצים את התלמידים להתמודד עם נושא החלוקה והעברת המידע הם מתקשים מאד. ממחקר שעסק במנגנון העברת פרמטרים עולה כי, אמנם, מרבית התלמידים מכירים את הכללים התחביריים הדרושים להעברת פרמטרים ואף יודעים להסביר את ההבדלים הסמנטיים בין פרמטרים של ערך לפרמטרים של משתנה, אך למעשה הם אינם מבינים לעומק את המנגנון. שכן, בבואם להתמודד עם בעיות תכנות הדורשות הבנה מעמיקה (כגון רקורסיה), התגלו קשיים רבים ותפיסות מוטעות (סולומונוביץ, 2003).

פתרון

- יש ללמד את תהליך הפירוק לתת משימות כבר בהתחלה. הלימוד יתבצע הן באמצעות מספר רב של דוגמאות פתורות שיוצגו לתלמידים והן באמצעות מתן מטלות לפתרון בעיות בשיטה זו.
- יש להכין תרגילים מורכבים במ מתבקש או הכרחי השימוש בתבנית ולבצע סימולציה של פתוח תוכנית מורכבת.
- חובה להדגיש כי השלב הראשון בפיתרון הבעיה הוא התכנון: חלוקה למשימות ותיאור החישובים ללא ההקפדה על התחביר של שפת התכנות.
- על המורה להיות עקבי, כלומר לפתור כל בעיה שמוצגת לתלמידים בשיטת החלוקה לתת משימות.
- יש ללוות את תהליך הלמידה בכלי של טבלאות מעקב ככלל ובעת לימוד נושא העברת פרמטרים בפרט.
- יש לבדוק את הבנת מנגנון העברת הפרמטרים באמצעות תרגילים מתאימים, לא שגרתיים (סולומונוביץ, 2003).

דיון / מסקנות / יישום

בתחילה, התלמידים לא מכירים בחשיבות חלוקת בעיה נתונה לתת משימות, אולם בכל זאת יש להקפיד על כך שכן, מתוך הבעיות שיוצגו וידונו בכיתה, עשויה לצמוח תבנית זו באופן טבעי ואינטואיטיבי. לאחר שהתלמידים יכירו בחשיבות התבנית, הם יצליחו יותר ביישומה. לפיכך, לימוד הטכניקה של פירוק לתת משימות הינו שלב הכרחי וחשוב בפתרון בעיות במדעי המחשב. כפי שצוין לעיל, התלמיד יסגל לעצמו הרגלים נאותים, אותם יישם בשלב מאוחר יותר. כמו כן, מניסיונו בעבר, במהלך ביצוע סימולציה של פיתוח תכנות מורכבות, התלמידים נוטים לבצע טעויות רבות, במיוחד בשלבים הראשונים. דיון בטעויות וניתוחן עשוי למנוע הישנותן בהמשך. זאת ועוד, חשוב מאד שלתלמידים תהיה אפשרות לדון בכיתה בנושאים מעניינים כגון: פתרונות מקוריים, התלבטויות וכו', אשר יעלו בזמן הסימולציה.

חשוב להדגיש את שלב תכנון הפיתרון (החלוקה לתת משימות והעברת המידע), ולא את שלב יישום הבעיה בשפת התכנות. זאת, מאחר ואחת מהמטרות של פיתרון בעיות במדעי המחשב, היא לימוד עקרונות תכנות נכונים ללא תלות בשפת התכנות.

טבלת מעקב היא כלי עזר חיוני ביותר לתלמידים. היא עוזרת להם לעקוב אחר השינויים בערכם של משתנים, פרמטרים בתוכנית ומאפשרת בכך לתלמידים ליצור מודל נכון של מנגנון זרימת המידע. המטרה בחשיפת תלמידים לשאלות מתאימות לא שגרתיות היא לעזור להם להבין טוב יותר את התהליך שקורה בזיכרון המחשב, וכיצד באמת פועל מנגנון העברת פרמטרים (סולומונוביץ, 2003).

משאבים מיוחדים

כל בעיה הניתנת לפיתרון באמצעות כתיבה מודולארית – מתאימה (החל מאלגוריתמים פשוטים וכלה באלגוריתמים מורכבים יותר).

על המורה לתכנן תרגילים ומשימות בדרגת קושי עולה, כך שבסוף תהליך ההוראה יוכלו התלמידים להתמודד עם תרגילים מורכבים, בהם השימוש בתבנית זו הכרחי.

קישור לתבניות נוספות

על פי התבנית *Early Bird* יש ללמד את הרעיונות המרכזיים והחשובים בהתחלה ואח"כ במהלך ההוראה לחזור עליהם כפי שהומלץ לעיל.

על פי התבנית *Toy box* יש לאפשר לתלמידים "לשחק" עם כלים מוכנים. בתבנית זו הכלים הם: טבלת מעקב ושגרות ספרייה.

על פי התבנית *Larger than life* יש ללמד עקרונות חשובים כבר בהתחלה (ללא ירידה לפרטים), למרות שהתלמידים עדין לא מכירים בחשיבות עקרונות אלה. כפי שצוין לעיל, יש להקפיד על פיתרון בעיות על ידי פירוקן לתת משימות כבר בשלבים המוקדמים של ההוראה.

על פי התבנית *Mistake* התלמיד מתבקש ליצור טעויות מסוימות ולאחר מכן לחקור את תוצאותיהן ועל פי התבנית *Grade It Again Sam* יש לספק לתלמיד סביבה בה הוא יכול לטעות ללא חשש וללמוד מטעויות אלה. תבנית *זרימת המידע* יכולה לגרום ל*Mistake* ול-*Grade It Again Sam* לעבוד ביחד בהצלחה.

דוגמאות לשימוש בתבנית

ניתן להשתמש בתבנית זו בהוראת תכנות מובנה בשפות כגון: פסקל, תכנות לוגי, ויזואל בייסיק, שפת C תכנות פונקציונאלי ועוד. בשלבים הראשונים מתבצעת חלוקה לתת משימות ללא זרימת מידע בין משימה למשימה. בשלבים מאוחרים יותר, כאשר השימוש בתבנית זו הכרחי, יש ללמד את מנגנון העברת הפרמטרים כאמצעי המאפשר תהליך של זרימת מידע.

חשוב ללוות את הלמידה בכלי עזר מתאימים במטרה לאפשר לתלמידים ליצור מודל נכון של מנגנון זרימת המידע בין השגרות.

מקורות

Bergin, J. (2002), "*Fourteen Pedagogical Patterns*", Pace University, N.Y. USA.

<http://csis.pace.edu/~bergin>

המחלקה להוראת המדעים (1996), יסודות מדעי המחשב – יסודות 1, הוצ' מכון ויצמן למדע, רחובות.
המחלקה להוראת המדעים (1997), יסודות מדעי המחשב – יסודות 2, הוצ' מכון ויצמן למדע, רחובות.
המרכז להוראת המדעים (1997), עיצוב תכנה, הוצ' האוניברסיטה העברית, ירושלים.
מן, ש. (2002), יסודות מדעי המחשב אלגוריתמיקה ושפת פסקל, חלק ב', הוצ' מבט לחלונות, הוד השרון.

סלומוביץ, מ. (2003), "האם מנגנון העברת הפרמטרים בתכנות אכן מוכן לתלמידים על בוריה?", הבטים בהוראת מדעי המחשב, ינואר 2003, עמ' 18 – 22.

מערך שיעור ופעילויות ליישום התבנית "זרימת מידע" ותבנית פדגוגית על פי Bergin

מיכל בראל וזהבה לוי

במערכי השיעור הבאים ובפעילויות הנלוות אנו מיישמות הן את התבנית הפדגוגית החדשה "זרימת מידע" והן את התבניות הפדגוגיות *Grade It Again Sam* ו- *Mistake* על פי Joseph Bergin.

משך הפעילות 2-3 שיעורים ויתכן אף יותר בהתאם לרמת התלמידים.
במהלך הפעילות, התלמידים נדרשים לחלק בעיה מורכבת לתת משימות ולנהל נכון את זרימת המידע ביניהן. בשיעורים משולבת עבודה עצמית של התלמידים, המלווה ברישום על שקפים. מניסיונו בעבר עולה כי מרבית התלמידים שוגים בחלק זה של הפעילות. לפיכך, על אף הטעויות הרבות, על המורה ליצור סביבה תומכת כך שהאווירה בכיתה תהיה נעימה והתלמידים יחשו כי הם רשאים לטעות (*Grade It Again Sam*).

בדיון שיערך בכיתה לאחר מכן התלמידים לומדים הן משגיאותיהם והן משגיאות חבריהם (*Mistake*). לדעתנו, שיעורים אלו יעילים יותר משיעורים פרונטאליים. הלמידה היא למידה פעילה ומהנה והתלמידים יוצאים נשכרים.

לאחר הפעילות העצמית מומלץ לערוך בכיתה דיון בנושאים כגון: עבודת צוות, יתרונות החלוקה למודולים עצמאיים, תהליך זרימת המידע ועוד. במהלך הדיון ניתן יהיה להרחיב ולדון בשאלות נוספות כמו: מה עדיף פרוצדורה או פונקציה? האם יש הבדל ביניהן? אם כן, מה עלינו לשנות עקב הבחירה? מי צריך לשנות? האם לשנות את כל התוכנית או רק חלק ממנה? האם חובה לקרוא לפרמטרים בשגרות השונות בשמות זהים?

חשוב מאד לעורר את מודעות התלמידים לשוני בתהליך זרימת המידע באם הם בוחרים במימוש האלגוריתם ע"י פרוצדורה או ע"י פונקציה. בנוסף, עולות השאלות: האם על השגרה להעביר מידע לשגרה אחרת או לתוכנית הראשית? האם כל המידע זורם אך ורק לתוכנית הראשית? מובן שניתן להרחיב את הדיון לכיוונים נוספים בהתאם לרמת התלמידים ולשאלות שהם מעלים.

העיתוי

לאחר שנלמדו הנושאים הבאים: פירוק למשימות ופרוצדורות, פרמטרים מטיפוס מערך, מציאת הערך הגדול/הקטן ביותר במערך ואיברי מערך כצוברים. לפי פרקים 1, 2 ו-3 עד סעיף 3.4 (לא כולל סעיף 3.3), בספר הלימוד: יסודות מדעי המחשב, יסודות 2.

מהלך השיעור

1. חלוקת דף בו מוצגת הבעיה הבאה:

מנהל חנות הספרים "ספרוני" החליט לערוך מעקב אחר 200 הספרים שבחנותו באמצעות המחשב. לשם כך ניתן מספר סידורי בין 1 ל-200 לכל ספר בחנות. בתחילת החודש הוקלדה הכמות במלאי של כל אחד מהספרים. במהלך החודש, בכל סוף יום, נרשמו עבור כל ספר שנמכר שני מספרים: המספר הסידורי של הספר ומספר העותקים שנמכרו ממנו באותו יום.

בעל החנות מעוניין לדעת בסוף החודש:

✓ מספר/י הספר/הספרים שאזלו.

✓ הספר/הספרים שנמצאים בחנות במספר עותקים מרבי.

פתח וישם אלגוריתם אשר הקלט שלו הוא:

(א) המלאי בתחילת החודש.

(ב) זוגות מספרים המתארים את מכירות הספרים בחנות. הזוג $(0,0)$ מציין את סוף הקלט.

פלט האלגוריתם הוא המידע הדרוש למנהל החנות.

הערה: הקפד שכל שלב באלגוריתם יבוצע ע"י פרוצדורה/פונקציה.

הערה: בשיעורים אלה, ניתן להציג לתלמידים כל בעיה מורכבת אחרת, מאותו סוג של הבעיה המוצגת לעיל.

2. דיון ראשוני (בשיתוף התלמידים) בדרכים לפתרון הבעיה כאשר הדגש הוא על קביעת קבועים, טיפוסים ומשתנים עיקריים (תיאור תפקידם וטיפוסם), הדרושים לפתרון. במהלך דיון זה נתמקד בשאלות: האם יש צורך במערך לפתרון הבעיה ומדוע? אם כן, מהו סוג המערך הדרוש: מערך צוברים, מערך מונים או מערך לשמירת נתונים? כיצד יש להגדירו?

3. חלוקת הבעיה לתת משימות (בשיתוף התלמידים), כך שכל תת משימה תמומש באמצעות שגרה מתאימה. כמו כן, יש לדון בקביעה עבור כל שגרה האם תיכתב כפרוצדורה או כפונקציה? בכיתות חלשות ניתן להוסיף ולדון בסוגי הלולאות הדרושות בשגרות השונות, אם בכלל.
4. חלוקת הכיתה לקבוצות. כל קבוצה אחראית לכתיבת שגרה בשפת התכנות הנלמדת, בצרוף טענות כניסה/יציאה מתאימות. קבוצה נוספת תהיה אחראית לכתיבת התכנית הראשית. כל אחת מהקבוצות מקבלת שקף עליו תרשום את השגרה לה היא אחראית. הקבוצה שאחראית לכתיבת התוכנית הראשית מקבלת שני שקפים: האחד לכתיבת כותרת התוכנית, כולל טענות קלט/פלט, קבועים, הגדרת טיפוסים והצהרה על המשתנים העיקריים. השקף השני נועד לכתיבת התוכנית הראשית.
- כל הקבוצות נדרשות לעבוד באופן עצמאי** ונאסר על חברי קבוצה אחת לדבר או להיעזר בחברי הקבוצות האחרות. בשלב זה על המורה לפקח שכל קבוצה מבצעת את המטלה לה נדרשה ואל לו לשאת פעולה או להשיב לשאלות התלמידים. רק חברי הקבוצה האחראית לכתיבת התוכנית הראשית, רשאים לעבור בין הקבוצות ולקבל מידם את המידע ההכרחי לביצוע משימתם, כלומר את כותרות השגרות.
5. איסוף השקפים והצגתם. יש להתייחס לכל שקף בנפרד ולתקן את השגיאות. כדאי להעלות את השאלה: האם יש חשיבות לסדר בו מוצגים השקפים ומדוע? יש לשוחח על התהליך שהתרחש בכיתה: החלוקה לצוותים והעבודה העצמאית של כל צוות. בנוסף, כפי שצוין לעיל, יש להרחיב את הדיון בשאלות כגון: שמות השגרות- על מי מוטלת האחריות? שמות הפרמטרים – האם יש לקרוא באותם שמות בכל השגרות? תפקיד הפרמטרים? מנגנון העברת הפרמטרים? סוג הפרמטרים הדרוש: פרמטר של ערך או פרמטר של משתנה? מי קובע? **לדעתנו שלב זה, בו התלמידים לומדים משגיאותיהם, הוא שלב חשוב ביותר בתהליך הלמידה.**
6. בשיעור הבא מומלץ לחלק לתלמידים פתרון הבעיה (הקוד בשפת התכנות) המבוסס על זה שהוצג במהלך הפעילות בכיתה. בפתרון זה יהיה שימוש באותם שמות של שגרות ופרמטרים שהתלמידים בחרו. חשוב להדגיש שאין הכוונה לחלוקת פתרון שהוכן מראש ע"י המורה.
7. סיכום השיעור והפעילות.
8. שעורי בית: התלמידים יחברו בעיה נוספת, הדומה לבעיה שנדונה בכיתה. הם יגישו את הבעיה ופתרונה על פי השלבים שהוצגו בכיתה:
- הצהרה על קבועים, טיפוסים ומשתנים כללים.
 - חלוקה לתת משימות, כך שכל תת משימה תמומש באמצעות שגרה מתאימה.
 - מימוש בשפת התכנות.
- התלמידים רשאים להגיש את המטלה בקבוצות של 2-3 תלמידים בכל קבוצה.

רשימת מקורות מומלצים

- Alexander, C., Ishikawa, S., & Silverstein, M. (1977). *A Pattern Language*, Oxford University Press.
- Astrachan, O. Berry, G., Cox, L., & Mitchener, G. (1998). Design patterns: An essential component of CS curricula. *SIGCSE Bulletin*, 30(1), 153-160.
- Bergin, J., Proulx, V.K., Brady, A.F., Hartley, S., Stockton, R., Kelemen, C., Klassner, F., Kumar, A., McNally, M., Mutchler, D., Rasala, R., Ross, R. (1999). Resources for Next generation introductory CS course: report of the ITiCSE'99 working group on resources for the next generation CS 1 course. *SIGCSE Bulletin*, 31(4), 101-105.
- Bergin, J. (2001). A pattern language for initial course design. *SIGCSE Bulletin*, 33(1), 282-286.
- Bergin, J. (2002a). *Fourteen Pedagogical patterns for teaching computer science*. Available: <http://csis.pace.edu/~bergin/PedPat1.3.html>
- Bergin, J. (2002b). *A Pattern Language for Course Development in Computer Science*. Available: <http://csis.pace.edu/~bergin/patterns/coursepatternlanguage.htm>
- Bergin, J., Brady, A., Duvall, R., Proulx, V., & Rassala, R. (2001). Using patterns in the classroom. *Journal of Consortium for Computing in Small Colleges*, 16(4), 5-7.
- Eckstein, J. (2000). Learning to teach and learning to learn: Running a course. In *Proceedings of EuroPLoP 2000*.
- Eckstein, J. (2001). Pedagogical patterns: capturing best practice in teaching object technology. *Software Focus*, 2(1), 9-12.
- Eckstein, J., Bergin, J., Sharp, H. (2002). Patterns for active learning. In *Proceedings of EuroPLoP'02*.
- Eckstein, J., Manns, M.L , Sharp, H., & Sipos, M. (2003). Teaching from different perspectives, submitted to *EuroPLoP'03*.
- Fincher, S. (1999). Analysis of design: an exploration of patterns languages for pedagogy. *Journal of Computers in Mathematics and Science Teaching: Special Issue CS-ED Research*, 18(3), 331-348.

Fincher, S. & Utting, I. (2002). Pedagogical patterns: their place in the genre. *SIGCSE Bulletin*, 34(3), 199-202.

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

Haberman, B. (2002). Frames and boxes: A pattern-based method for manipulated binary trees. *SIGCSE Bulletin*, 34(4), 60-64.

Harrison, N.B. (1999). The language of shepherding a pattern for shepherds and sheep. In *Proceedings of PloP'99*.

Jalloul, G. (200). Links: a framework for object-oriented software engineering. *Computer Science Education*, 10(1), 75-93.

Lilly, S. (1996). Patterns for pedagogy. *Object Magazine*, 5(8), 93-96.

Linn, M.C., & Clancy, M.J. (1992). The case for case studies of programming problems. *Communications of the ACM*, 35(3), 121-132.

Muller, O., Haberman, B., & Averbuch, H. (2004). (an Almost) Pedagogical pattern for pattern-based problem-solving instruction, *ACM SIGCSE Bulletin*, 36(3), 102-106.

Reed, D. (1998). Incorporating problem-solving patterns in CS1. *SIGCSE Bulletin*, 20(1), 6-9.

Riehle, D. & Zullighoven, H. (1996). Understanding and Using Patterns in Software Development. *Theory and Practice of Object Systems*, 2(1), 3-13.

Sharp, H., Manns, M.L., & Eckstein, J. (2003). Evolving pedagogical patterns: the work of the Pedagogical Patterns Project, *Computer Science Education*, 13(4), 315-330.

Wallingford, E. (2000). Using patterns in the CS curriculum. *Journal of Computing in Small Colleges*, JCSC, 15(5), 238-240.

<http://www.cs.uni.edu/~wallingf/patterns/elementary/ccscne00/presentation.pdf>

עבודה הדנה בתרומת תבניות תכנות ותבניות תכנון לתהליכי למידה

מסגרות לתבניות פדגוגיות (forms)

אתר הבית של פרויקט תבניות פדגוגיות

האתר של סאלי פינצ'ר