

תוכן הגליון

מאמר המערכת.....	2
כנס תשס"ה וקורס מורים מובילים תשס"ה	
פרטים והרשמה	4
ניתוח שגיאות וקשיי תלמידים בבחינות הבגרות: שאלון עיצוב תכנה	
רונית בן-בסט לוי, ד"ר ברוריה הברמן	7
בניית משחק בפרדיגמה הפונקציונלית: התנסות בתהליך של פיתוח תכנה	
תמר פז, ארנה מילר	12
הוראת מדעי המחשב כאזורים פוריים של מפגש תרבויות	
יפעת בן-דוד קוליקנט, מוטי בן-ארי	21
תכנות מונחה עצמים עם תחילת המאה ה- 21, סקירת ספרות, חלק ראשון	
עפרה ברנדס	30
שימוש בחידות כאמצעי לשיפור מיומנויות של פתרון בעיות אלגוריתמיות	
שרה פולק, צביקה פירסט	39
פיצוחים: משחק לימודי בנושא מבנה המחשב	
אהובה שפרלינג, דורית ליקרמן	45
תורת המחשב: חומרים מסדנה פדגוגית של המרכז הארצי	
זיוה קונצמן	49
יסודות מדעי המחשב: חומרים מסדנה פדגוגית של המרכז הארצי	
פריידה בן-פזי	56
ניתוח מערכות מידע: הצעות לארועים ופרויקטים	
מורים מובילים	61
תבל: מחשב ומחשבה	68
טופס משוב	69

מאמר מערכת

קוראים יקרים

לפניכם הגליון השני של "הבטים בהוראת מדעי המחשב" לשנת תשס"ד. גם הגליון הנוכחי נשלח לבתי הספר העל-יסודיים ללא תשלום, הודות למדיניותו ותמיכתו של מטה מל"מ - המרכז הישראלי לחינוך מדעי טכנולוגי. אם אתם מעוניינים להמשיך ולקבל את העיתון ללא תשלום, אנא מלאו את טופס המשוב המופיע בעמוד האחרון ושלחו אותו למטה מל"מ.

ומה בגליון הפעם?

תכנית הלימודים במדעי המחשב מורכבת מיחידות שונות ומגוונות. בגליון הנוכחי ניסינו לתת ביטוי למספר גדול ככל האפשר של תחומים מתכנית הלימודים. תוכלו למצוא מאמרים העוסקים ביחידות החובה יסודות מדעי המחשב ועיצוב תכנה, אבל גם מאמרים וחומרים הקשורים ליחידות הבחירה תכנות פונקציונלי, מערכות מידע, תורת המחשב, חישוב מקבילי ומבוזר, ותכנות מונחה עצמים. בנוסף, תוכלו למצוא מספר מאמרים העוסקים בשיטות הוראה כלליות, כמו חידות ומשחקים.

בגליון הקודם פרסמנו ניתוח של שאלוני בגרות ביסודות מדעי המחשב. הפעם נפרסם את חלקו השני של המחקר שנערך על-ידי מרכז המורים הארצי למדעי המחשב, ובו ניתוח **שגיאות אופייניות וקשיי תלמידים בבחינות הבגרות בעיצוב תכנה**. הניתוח נערך בהסתמך על תשובות תלמידים לשאלונים מבחינות קיץ תשס"ב.

עקב היקפו הנרחב של המחקר, לא נוכל להביא כאן את כל הממצאים והשלכותיהם. אתם מוזמנים לבקר באתר המרכז הארצי ולקרוא שם את הניתוח במלואו.

המאמר של תמר פז וארנה מילר עוסק **בהתנסות בתהליך של פיתוח תכנה** במסגרת לימוד היחידה "תכנות פונקציונלי". הן מתארות כיצד בניית הפרויקט מאפשרת לתלמידים להתנסות בתהליך השלם של פיתוח תוכנה מורכבת בזמן קצר יחסית, ולהגיע לתוצר שהוא, בדרך-כלל, תוכנה של משחק.

יפעת בן-דוד קוליקנט ומוטי בן-ארי מתמודדים עם קשיי למידה בחינוך מדעי המחשב מזווית מיוחדת. הם מתמקדים בהתנגשות בין תרבות המחשבים הלא פורמאלית של תלמידים לבין תרבויות מדעי המחשב האקדמאית והמקצועית. כפתרון, הם מציעים גישה הנקראת **אזורים פוריים של מפגש תרבויות**. הגישה הזו מוצעת כדרך לנתח את קשיי הלמידה של התלמידים וכפתרון לשיפור משמעותי של הלמידה. הגישה הפדגוגית שלהם מודגמת על ידי ניתוח אפיזודות מהקורס "חישוב מקבילי ומבוזר".

עפרה ברנדס מגישה לנו סקירת ספרות בנושא **תכנות מונחה עצמים עם תחילת המאה ה-21**. בסקירה הזו, שהיא חלק מעבודת המסטר שלה, עפרה מעמתת את הגישות המציעות להתחיל את הוראת מדעי המחשב בתכנות פרוצדורלי לעומת הגישות התומכות להתחיל מתכנות מונחה עצמים. עקב אורכה של הסקירה, יפורסם הפעם רק חלקה הראשון המציג את המושגים הנידונים ואת הבעיה העומדת על הפרק. כפי שעפרה מציינת, נשמח אם פרסום הסקירה יעודד דיונים ענייניים בנושא, ואם יובאו לתשומת לב כולנו מחקרים ומאמרים חדשניים בתחום.

למען המורים שלא השתתפו בסדנאות וכדי שתוכלו להתרשם מהחומרים האלה, בחרנו להביא לכם הפעם שתי דוגמאות: **חומרים ביסודות מדעי המחשב** שהוכנו על-ידי פרידה בן פזי לסדנה בירושלים ו**חומרים בתורת המחשב** שהוכנו על-ידי זיוה קונצמן לסדנה בתל אביב. אתם מוזמנים להיכנס לאתר האינטרנט של המרכז ולהעזר גם בחומרים שהוכנו לסדנאות האחרות.

כמו תמיד, תוכלו למצוא בגליון הודעות על פעילויות שונות של מרכז המורים הארצי. במיוחד אנחנו מבקשים להסב את תשומת ליבכם **לקול קורא להגשת הצעות לכנס המורים הארצי** שיתקיים בחנוכה תשס"ה, ולפרטים על **קורס מורים מובילים** תשס"ה.

שימו לב: אנחנו נמצאים בעיצומן של ההכנות לקראת השנה הבאה - תשס"ה. נשמח לשמוע מכם למה אתם זקוקים ואיך נוכל לסייע לכם בעבודה. רוב הסדנאות שהתקיימו השנה אורגנו במענה לפניית כאלה שקיבלנו מכל הארץ. תוכלו לפנות אלינו בטלפון, בפקס, או בדוא"ל. כל הפרטים מופיעים על כריכת העיתון.

כתב העת מופיע במלואו גם באתר האינטרנט של המרכז הארצי <http://cse.proj.ac.il>. הכניסה לאתר והשימוש בו אינם כרוכים בתשלום. אתם מוזמנים לגלוש, לקרוא, או להוריד קבצים. לאחר שתקראו את העיתון, אתם מוזמנים למלא את טופס המשוב ולשלוח אותו למטה מל"מ.

נשמח לקבל מכם חומרים נוספים מפרי עטכם. אתם יכולים לשלוח אלינו מאמרים, הצעות לפעילויות לימודיות בכיתה או במעבדה, גישות שונות להוראה, לבטים הנוגעים להוראת מדעי המחשב, או כל נושא אחר שלדעתכם יעניין את קוראי העיתון.

קריאה מהנה וחופש נעים,

ממערכת העיתון וצוות המרכז הארצי

לפני מספר חודשים הוספנו לאתר האינטרנט של המרכז הארצי פינה חדשה שמוקדשת לחידות. בהמשך לפעילות שהתקיימה בפינה הזו ובפורום המלווה אותה, מציעים שרה פולק וצביקה פירסט דרך הוראה מרעננת: **שימוש בחידות כאמצעי לשיפור מיומנויות של פתרון בעיות אלגוריתמיות**. הם מציינים תהליך של פתרון בעיות, אסטרטגיות מקובלות לפתרון חידות ומסיימים עם דוגמה לשימוש בחידה בהוראת יסודות מדעי המחשב.

בגליונות הראשונים של "הבטים" הבאנו משחקים לימודיים בנושאים שונים. דורית ליקרמן ואהובה שפרלינג מציעות הפעם **משחק לימודי** (במתכונת פיצוחים) **בנושא מבנה המחשב**. נשמח לקבל מהקוראים הצעות נוספות למשחקים.

אתר האינטרנט של המרכז הארצי צובר תנופה ומכיל כבר היום חומרים רבים שיכולים לסייע לכם. כדי לסייע לגולשים למצוא את החומרים ביתר קלות בנינו **קטלוג** שמכיל קטגוריות שימושיות. הקטלוג הזה מתווסף לפורטל יחידות הלימוד שמאפשר לכם גישה לחומרים לפי תחומי תכנית הלימודים. כדי לעודד אתכם להעזר בקטלוג ובפורטל יחידות הלימוד בחרנו להביא דוגמה מהחומרים של מבוא למערכות מידע ובשתי דוגמאות מחומרים שפותחו על-ידי מורים מובילים לסדנאות שהתקיימו השנה.

מורים מובילים בתחום **מבוא למערכות מידע**, בהנחיית שרה פולק וצביקה פירסט פיתחו הצעות לארגון למידה באמצעות **פרויקט מתפתח**, והצעות מגוונות **לארועים ופרויקטים**. בגליון הנוכחי תוכלו למצוא מבחר קטן של החומרים האלה.

במהלך השנה ארגן המרכז הארצי כ- 30 **סדנאות פדגוגיות במדעי המחשב** בהנחיית מורים מובילים מהשורה הראשונה. הסדנאות האלה התקיימו בכל הארץ והוקדשו להבטים פדגוגיים שונים של תכנית הלימודים. רוב החומרים שחולקו במסגרת הסדנאות עומדים לרשותכם באתר המרכז הארצי.

קול קורא להגשת הצעות לכנס המורים הארצי תשס"ה

הקדמה

המטרה העיקרית של כנס המורים השנתי היא לספק הזדמנות למפגש חברתי ומקצועי, לאפשר למורים לחלוק את המומחיות המקצועית שלהם עם עמיתים למקצוע, להתחבט בצוותא בשאלות מקצועיות, ולעודד שיתוף פעולה חברתי ומקצועי בתוך הקהילה של מורי מדעי המחשב וטכנולוגיות מידע.

בנוסף להרצאות מוזמנות ותערוכת חומרי לימוד ינתן בכנס מקום נכבד להצגת הצעות שיוגשו על ידי מורים. ההצגה תוכל להתבצע במסגרת הרצאות קצרות, סדנאות, דיוני שולחן עגול ופוסטרים בכל אחד מנושאי הכנס.

הכנס יתקיים בחופשת חנוכה תשס"ה. על המועד והמקום תפורסם הודעה נפרדת.

אנו מזמינים את כל העוסקים בהוראת מדעי המחשב וטכנולוגיות מידע להגיש הצעות לכנס. ניתן להגיש הצעה בכל אחד מהנושאים המופיעים בהמשך. **מועד אחרון להגשה 10.10.04**

מחבר של הצעה שהתקבלה חייב להציגה באופן אישי. אם אינו יכול לעשות זאת, עליו ליצור קשר עם הועדה המארגנת כדי לתאם מחליף. שימו לב: רק המחבר הראשון ישובץ כך שמושביו לא יתנגש עם מושבים אחרים שלו. המציג יהיה חייב להירשם לכנס. הצעה שלא תעמוד בהנחיות ההגשה, תידחה על הסף.

נושאי הכנס

ניתן להגיש הצעות לכנס בכל אחד מהנושאים הבאים: (אפשר לקבל רעיונות גם באתרי הכנסים הקודמים)

1. רעיונות הוראה.
2. מחקר בתחום למידה/הוראה של מדעי המחשב וטכנולוגיות מידע.
3. תכנון לימודים של יחידה או תחום תוכן, תכנון כיתתי או בית ספרי, מסגרות למידה מיוחדות (למשל, חוגי הורים ותלמידים).
4. שיטות הוראה: שיטות הוראה חלופיות, סגנונות הוראה.
5. הערכת השגים של תלמידים: שיטות הערכה חלופיות, הערכה עצמית של לומדים, הקשר בין הערכה ללמידה והוראה.
6. תהליכים קוגניטיביים: יכולת ואינטליגנציה, שונות של לומדים, תהליכי חשיבה של תלמידים.
7. סביבות למידה: סביבות שיתופיות, בינתחומיות, מתוקשבות.
8. הוראת מדעי המחשב וטכנולוגיות מידע לאוכלוסיות מיוחדות (למשל, מחוננים).
9. הוראת מדעי המחשב וטכנולוגיות מידע במוסדות על-תיכוניים (למשל, מכללות להכשרת מורים).
10. הכשרת מורים: הכשרת פרחי הוראה, השתלמויות מורים.

היכונו ... היכונו ... היכונו ... היכונו ... היכונו ...

גם בשנת תשס"ה יום הפעילות המרכזי יהיה יום רביעי.

בתחילת ספטמבר ישלח לכל בתי הספר מכתב הערכות ובו יופיעו הפרטים על

הפעילויות המתוכננות. אתם מוזמנים גם לעקוב אחר הפרסומים השוטפים באתר המרכז הארצי ובאתר הפיקוח.

הגשת הצעות לכנס

ניתן להגיש הצעות להרצאה (20-30 דקות), סדנה (30-60 דקות), שולחן עגול (30-60 דקות), או פוסטר בכל אחד מנושאי הכנס. ההנחיות להגשת ההצעות מופיעות בהמשך. כל ההצעות שתתקבלנה תפורסמנה בחוברת הכנס.

מה צריך להגיש? (עד לתאריך 10.10.04)

• **ההצעה:** תוגש בעברית ותהיה בהיקף של עד 1,000 מילים (להוציא מקורות ביבליוגרפיים), בגופן (פונט) David או Times New Roman, בגודל 12, על נייר A4 (8.5" x 11"), רווח של שורה וחצי. בשורה הראשונה יופיע שם ההצעה ממורכז ומודגש. בשורה השנייה, שם/שמות המחבר(ים). שם המרצה המציג יסומן בקו. בשורה שלישית כתובת דוא"ל.

• **טופס נלווה להגשת הצעה** – יש להגיש לכל הצעה בנפרד.

• **אופן ההגשה:** רצוי לשלוח את ההצעה והטופס הנלווה בדואר אלקטרוני לכתובת lapidot@tx.technion.ac.il

אם אין לכם גישה לדוא"ל,
נא לשלוח את ההצעה והטופס הנלווה
לפקס 04-8293004

טופס נלווה להגשת הצעה לכנס

נבקשך למלא טופס זה ולצרף אותו אל ההצעה.

תאריך אחרון לקבלת החומר: 10.10.04

שם המרצה:

כתובת:

טלפון:

טלפון נייד:

דוא"ל:

מקום עבודה:

כותרת ההצעה:

מחברים נוספים:

ההצעה מוגשת להצגה במסגרת הבאה:

(נא לסמן בתיבה המתאימה)

[] מושב הרצאות קצרות [] סדנה

[] דיון שולחן עגול [] הצגת פוסטר

נא לסמן את התחום המתאים ביותר להצעה:

תחום	עזרים
1 רעיונות הוראה	(נא לסמן כל מה שדרוש): [] לוח [] מטול שקפים [] מחשב [] מקרן למחשב [] אחר (פרט)
2 מחקר למידה/הוראה מדהמ"ח וטכ"מ	
3 תכנון לימודים	
4 שיטות הוראה	
5 הערכת השגים של תלמידים	
6 תהליכים קוגניטיביים	
7 סביבות למידה	
8 אוכלוסיות מיוחדות	
9 מוסדות על תיכונים	
10 הכשרת מורים	

במידה וההצעה תתקבל, אני מתחייב/ת להציגה בכנס.

חתימה

תאריך

קורס מורים מובילים במדעי המחשב – תשס"ה

טופס הרשמה לקורס מובילים תשס"ה

נא להחזיר למרכז המורים הארצי למדעי המחשב,
המחלקה להוראת הטכנולוגיה והמדעים,
טכניון, חיפה 32000, פקס 04-8293004

סיום ההרשמה לקורס: יום ראשון 12.9.04

שם:

ת.ז.:

כתובת:

טלפון:

נייד:

דוא"ל (חובה):

בית ספר:

יישוב:

מלמד מדעי המחשב _____ שנים.

הגשתי תלמידים לבגרות ביחידות הבאות:
(פרט את שמות היחידות וכמה שנים הגשתי בכל אחת מהן)

יש לי רקע בתכנות מונחה עצמים (פרט):

הערות נוספות:

(נא לפרט שיקולים נוספים שלדעתך ראוי שנתחשב בהם
בקבלתך לקורס)

אנו שמחים להודיע על פתיחת ההרשמה לקורס מורים
מובילים במדעי המחשב לשנת תשס"ה.

הקורס יוקדש להכנה פדגוגית לקראת המעבר לשפות
התכנות החדשות במדעי המחשב, להעשרה ולהבטים
שונים של הובלת עמיתים.

בקבלה לקורס תינתן עדיפות למורים בעלי רקע
בתכנות מונחה עצמים.

הקורס מתאים גם למורים שהשתתפו בעבר בהשתלמויות
למורים מובילים.

הקורס יהיה בהיקף כולל של 112 שעות (מוכר לגמול
עם ציון) ויורכב מהפעילויות הבאות:

* **12 מפגשים** (72 שעות) - יתקיימו בתל אביב **פעם**

בשבועיים בימי רביעי בשעות 14:30 - 9:30

הקורס יפתח בסוף אוקטובר 2004.

* **ביצוע שני פרויקטים** (20 שעות כ"א)

בשלב הראשון של הקורס יוצעו מספר תחומים
אפשריים לפרויקט, כמו תמיכה מתוקשבת, הנחיית
פרויקטים כשיטת למידה, כתיבת מאמר, פרדיגמות
תכנותיות, תבניות פדגוגיות, ומחקר פעולה. כל
משתתף בקורס יבחר שני תחומים (בתאום עם צוות
הקורס) ויגיש פרויקט מסכם בהיקף של 20 שעות
בכל תחום. תתאפשר עבודה בקבוצות קטנות.
הפגישה האחרונה בקורס תוקדש להצגת
הפרויקטים.

* **השתתפות בכנס המורים הארצי תשס"ה**

התשלום לקורס יגבה רק מהמורים שיתקבלו לקורס.

מורים המעוניינים להשתתף בקורס מתבקשים למלא
את טופס ההרשמה המצורף

ולשלוח אותו אלינו **עד ליום ראשון 12.9.04.**

לפרטים נוספים אפשר לפנות אל המרכז הארצי

lapidot@tx.technion.ac.il

ניתוח שגיאות וקשיי תלמידים בבחינות הבגרות שאלון עיצוב תכנה – קיץ תשס"ב

רונית בן-בסט לוי, ד"ר ברוריה הברמן

מחשבה – מרכז המורים הארצי למדעי המחשב

תיאור השאלות לפי תחומי תוכן מתכנית הלימודים, סוג השאלה והמשימה לביצוע

משימות לביצוע	סוג השאלה	תחומי תוכן מתכנית הלימודים	
<ul style="list-style-type: none"> * התמודדות עם טיפוס נתונים מורכב * בחירת טיפוס הנתונים המתאים * ייצוג מלא של הפרטים המופיעים בשאלה * בניית ממשק עברי לכל פעולה * כתיבת שמות פעולות באופן פורמלי תוך הקפדה על פרמטרים, משמעות הפעולה והנחות 	<p>שאלת פיתוח : תוצר ממשק עברי וטיפוס נתונים מופשט</p>	<ul style="list-style-type: none"> * טיפוס נתונים מופשטים : הגדרה, ייצוג ושימוש בהם * עבודה עם טיפוס נתונים מופשט דרך ממשק עברי * מגבלות הייצוג של טיפוס נתונים מופשט. 	1
<ul style="list-style-type: none"> * הבנת הדוגמה (ללא הבנת הדוגמה קשה מאד לענות על השאלה) * שימוש נכון בממשקים של עץ בינארי ושל תור * התאמת אלגוריתם מוכר (סריקה לפי רמות המצוי בספר) לנדרש בשאלה * זיהוי רמה בעץ * בדיקת מיון של רמה 	<p>שאלת פיתוח : תוצר אלגוריתם</p>	<ul style="list-style-type: none"> * עבודה עם טנ"מ עץ דרך ממשק בסביבת עבודה * עץ בינארי : טיפול ברמה בעץ, זיהוי שורש עץ, תת-עץ (שמאלי וימני), סריקת עץ תור 	2
<ul style="list-style-type: none"> * הבנת השאלה * בחירת טיפוס עזר מתאימים * איתחול טיפוס העזר * מעבר על כל המוט עד לריקונו מטבעות ובמהלך המעבר סיווג הטבעות לאחד משני טיפוס העזר בהתאם לסוג הטבעת. * העברת הטבעות מטיפוס העזר חזרה למוט * ניתן היה לעבוד גם ללא טיפוס עזר תוך שמירת שני סוגי הטבעות ומניית כמות הטבעות מכל סוג ואז הכנסת טבעות למוט המקורי לפי כמות הטבעות מכל סוג. בכל מקרה חייבים לשמור את סוג הטבעת 	<p>שאלת פיתוח : כתיבת פרוצדורה בפסקל או ב-C</p>	<ul style="list-style-type: none"> * הכרת המושג "בסביבת העבודה" * הכרת טיפוס נתונים מופשט ושימוש בו, כאשר התלמיד אינו יודע כיצד מוגדר הטיפוס בפועל – עקרון הסתרת המידע * שימוש בממשק בסביבת עבודה * הפעלת פעולות הכתובות בממשק * עבודה עם טנ"מ דרך ממשק בסביבת עבודה 	3א

משימות לביצוע	סוג השאלה	תחומי תוכן מתכנית הלימודים	
<p>* חישוב פונקציית זמן ריצה של הפרוצדורה בהתאם לסדרי הגודל בשאלה ולסיבוכיות הפעולות על טיפוס העזר שהתלמיד בחר בהם</p> <p>* מתוך פונקציית זמן הריצה בנייה של הסיבוכיות תוך ציון מהות ה - n</p> <p>* לחילופין ניתן לתת הסבר מילולי – לולאות שבתוכן פעולות שכ"א מהן סדר הגודל שלה נתון בשאלה. יש לקחת בחשבון את סדר הגודל של הפעולות על-פי טיפוס העזר ולהתייחס למהות ה - n</p>	<p>שאלת פיתוח : כתיבת פרוצדורה בפסקל או ב - C</p>	<p>* סיבוכיות זמן ריצה – יעילות</p> <p>* הכרת המושג "סדר גודל"</p> <p>* ניתוח סיבוכיות זמן ריצה של אלגוריתם</p>	ב3
<p>* הבנה של האלגוריתמים מצא והחלף</p> <p>* בניית טבלת מעקב</p> <p>* הפרדה בין הקריאות הרקורסיביות</p> <p>* חישוב תוצאת הרצה בעבור קלט מסויים</p> <p>* ניסוח הבעיה אותה האלגוריתם פותר</p> <p>* ניסוח כללי קצה – טיפול בתחום שבין 1 לבין 7 וטיפול מ-8 ומעלה</p>	<p>שאלת מעקב וניתוח</p>	<p>* רקורסיה – קריאה רקורסיבית, מעקב אחרי אלגוריתם רקורסיבי</p> <p>* הבנה של אלגוריתם וניתוחו – ניסוח הבעיה אותה האלגוריתם פותר</p> <p>* הכללה של האלגוריתם</p> <p>* טיפול במקרי קצה</p>	4

באתר המרכז הארצי תוכלו למצוא הצעת "פתרון בית ספר" לכל אחת משאלות המבחן.

פתרונות שגויים וסיכום קשיים

שאלה מספר 1 סעיף א

שם הקבוצה: מחרוזת. שדה שני יהיה שחקנים שהוא מערך בגודל 10 המורכב ממספרים ממשיים שהוא הניקוד שצבר כל שחקן. שדה שלישי יהיה סה"כ: המכיל את סך כל הנקודות שצברה הקבוצה.

יום הספורט הוא מערך בגודל G שכל אחד מאיבריו הוא קבוצה המוגדרת למעלה.

3. התעלמות ממשקלי התחנות.

קבוצה: רשומה הבנויה משלושת השדות הבאים: שם הקבוצה: מחרוזת. שדה שני יהיה שחקנים שהוא מערך בגודל 10XK המורכב ממספרים ממשיים שהוא הניקוד שצבר כל שחקן בכל אחת מ-K התחנות. שדה שלישי יהיה סה"כ: המכיל את סך כל הנקודות שצברה הקבוצה.

יום הספורט יהיה רשומה המכילה בתוכה שני שדות:

1. אין אריזה של "יום-ספורט" כטיפוס נתונים אחד.

קבוצה: רשומה הבנויה משלושת השדות הבאים: שם הקבוצה: מחרוזת. שדה שני יהיה שחקנים שהוא מערך בגודל 10 המורכב ממספרים ממשיים שהוא הניקוד שצבר כל שחקן. שדה שלישי יהיה סה"כ: המכיל את סך כל הנקודות שצברה הקבוצה.

שדה משקלי התחנות - מערך בגודל K שכל אחד מאיבריו הוא מספר ממשי, בתחום שבין 0 לבין 1, המייצג את משקל התחנה.

שדה הקבוצות - מערך בגודל G שכל אחד מאיבריו הוא קבוצה המוגדרת למעלה.

2. התעלמות ממשקלי התחנות.

קבוצה: רשומה הבנויה משלושת השדות הבאים:

- שדה משקלי התחנות – מערך בגודל K שכל אחד מאיבריו הוא מספר ממשי, בתחום שבין 0 לבין 1, המייצג את משקל התחנה.
- שדה הקבוצות – מערך בגודל G שכל אחד מאיבריו הוא קבוצה המוגדרת למעלה.

שאלה מספר 1 סעיף ב

- 1. כתיבת אלגוריתמים במקום ממשק עברי הדפס מנצחים (DAY)
- 1. עבור I מ 1 ועד 3 בצע
- 1.1 הדפס `DAY.GROUP[I].NAME`
- הערה: התלמיד הניח כי 3 הקבוצות הראשונות מנצחות.

2. שימוש בפרמטר פלט בממשק

- אתחל יום-ספורט (DAY) – הפעולה מחזירה בפרמטר DAY יום ספורט שכל שדותיו מכילים אפסים או מחרוזות ריקות.
- מצא מנצחים (DAY,G1,G2,G3) – הפעולה מקבלת יום ספורט DAY ומחזירה בפרמטרים $G1, G2, G3$ את שמות הקבוצות המנצחות. הנחה: יום-ספורט DAY מאותחל.

3. אין שימוש בפרמטרים כלל

- הדפס מנצחים – הפעולה מקבלת יום ספורט DAY ומציגה כפלט את שמות הקבוצות המנצחות. הנחה: יום-ספורט DAY מאותחל.

4. לא מוזכרות הנחות

- מצא מנצחים (DAY) – הפעולה מקבלת יום ספורט DAY ומציגה כפלט את שמות הקבוצות המנצחות.

סיכום קשיים

- קושי במציאת ייצוג מתאים - למשל, תלמידים נוטים להגדיר באופן חלקי תוך התעלמות מחלק מפרטי השאלה, כמו התעלמות מקיום G קבוצות או קיום K תחנות. תלמידים לא הגדירו את משקל התחנה.
- קושי בהבנת המושג טיפוס נתונים מופשט המורכב ממספר תכונות - למשל, אין אריזה של יום הספורט לטיפוס נתונים אחד. התלמידים יצרו לחוד קבוצות ולחוד תחנות ולא ארזו אותם יחדיו.
- בכתובת הממשק העברי יש תלמידים שלא ביצעו פעולת איתחול או לחילופין לא ציינו בהנחות כי טיפוס הנתונים "יום-ספורט" מאותחל.
- חוסר בפירוט מתאים של הפעולות - למשל, בממשק העברי תלמידים ציינו שם פעולה בלבד מבלי לציין פרמטרים, היו תלמידים שהשתמשו בפרמטר פלט.
- בממשק העברי תלמידים לא ציינו הנחות.
- חוסר הבנת המושג "ממשק עברי" - למשל, תלמידים מבינים ממשק עברי ככתיבה של אלגוריתם. לכן בתשובתם נתנו אלגוריתמים מילוליים לכל פעולה במקום ממשק עברי.

שאלה מספר 2

- 1. התלמידים העתיקו את האלגוריתם לסריקה לפי רמות מהספר והסתפקו בכך.
- 2. התלמידים כתבו ללא תעוד – חסרות טענות הכניסה והיציאה.
- 3. אלגוריתם מבלי להתייחס לרמות פשוט השתמשו בסריקה רגילה של עץ.
- 4. אלגוריתם הדומה לדוגמה ולא החזירו אמת (בסעיף 5) או לא החזירו שקר.

סיכום קשיים

- אי ידיעת כללי כתיבת אלגוריתם – תלמידים לא כתבו כותרת ותיעוד לשאלה זו.
- התאמה שגויה – תלמידים זיהו כי מדובר בסריקה לפי רמות של עץ ובחרו להעתיק את האלגוריתם מהספר והסתפקו בכך. אין הדבר מספיק לצורך פתרון הבעיה.
- הכללה – תלמידים מורגלים לפתור בעיות בעצים בעזרת רקורסיה ובחרו לפתור גם בעיה זו כך - תלמידים אלה שגו בכך שהשוו רק שני בנים ולא רמה שלמה, או לא ביצעו השוואה כלל.
- תלמידים שהשתמשו ברקורסיה אינם יודעים להשתמש בה. למשל שכחו שיש לדאוג להחזרת אחד משני הערכים: אמת או שקר. האלגוריתם שלהם החזיר או אמת או שקר.
- אלה שהשתמשו בתור שכחו לאתחל את התור.

- חוסר בהבנת גודלו האינסופי של עץ- למשל, תלמידים לא בחרו בתור אלא במעריך. מעריך גודלו סופי. לכן לא ניתן להעביר אליו תוכן של רמה בעץ, שכן איננו יודעים מראש מהי כמות הנתונים ברמה. תלמידים אלה בדקו אם המעריך שנבנה ממויין.
- הבנת פעולות הממשק – למשל, תלמידים זימנו פעולות מהממשק מבלי להעביר אליהן פרמטרים ככתוב בממשק, או לחילופין זימנו פונקציה כפרוצדורה ולהפך.
- הבנת המושג בסביבת העבודה. למשל, תלמידים כתבו אלגוריתם ולא בפסקל או ב – C.

- הגדרת כותרת מתאימה – תלמידים הגדירו פונקציה – הגדרה שגויה שכן פונקציה בפסקל לא יכולה להחזיר טיפוס שאינו פשוט.
- תלמידים הגדירו מחדש את ה – Pole_type. הם הגדירו אותו כמחשנית כי זה "הזכיר להם מחשנית". הגדרה מחדש של טיפוס מוגדר מראש בשאלה היא שגויה.
- תלמידים הגדירו טיפוס עזר כמעריך – הדבר שגוי שכן כמות הטבעות על המוט אינה ידועה מראש.
- תלמידים שכחו לאתחל את טיפוס העזר.
- תלמידים שבחרו לספור את כמות הטבעות לא שמרו את סוג הטבעת ולכן לאחר ריקון המוט הנתון לא יכלו לבנות אותו מחדש לפי הסדר הדרוש.

שאלה מספר 3 ב

1. התלמיד מתעלם מסדרי הגודל שניתנו בשאלה סיבוכיות זמן הריצה היא $O(n^2)$ - מכיוון שיש פעולת הוצאה שהיא $O(n)$ והיא בתוך לולאה. פעולת ההוצאה בתרגיל היתה בסיבוכיות $O(1)$.
2. התלמיד עונה כפונקציה זמן ריצה סיבוכיות זמן הריצה היא $O(5n+7)$ התלמיד ספר את הפעולות לפי התכנית שכתב ולא התייחס לסדרי גודל.
3. התלמיד נתן מספר אקראי כלשהו כשובה. התלמיד ענה כי הסיבוכיות היא : 500 (למשל).

סיכום קשיים

- תלמידים לא התייחסו לסדרי הגודל של הפעולות בטיפוסי הנתונים שנבחרו כדי לחשב סיבוכיות.
- תלמידים התעלמו מסדרי הגודל שניתנו בממשק כנתונים בשאלה והמציאו סדרי גודל משל עצמם.
- תלמידים קובעים סיבוכיות מבלי לנמק או מבלי לציין את מהות ה – n.

- חוסר בהבנת גודלו האינסופי של עץ- למשל, תלמידים לא בחרו בתור אלא במעריך. מעריך גודלו סופי. לכן לא ניתן להעביר אליו תוכן של רמה בעץ, שכן איננו יודעים מראש מהי כמות הנתונים ברמה. תלמידים אלה בדקו אם המעריך שנבנה ממויין.

שאלה מספר 3 א

1. התלמיד הגדיר את טיפוס Pole_type מחדש.
Type Pole_type = stack_type;
2. התלמיד הגדיר פונקציה במקום פרוצדורה.
Function arrange_pole (P:Pole_type):
Pole_type;
3. התלמיד הגדיר טיפוס עזר כמערכים.
Procedure arrange_pole (var P:Pole_type);
Var Big, Small : array [1.. n] of Ring_type;
4. שכח לאתחל טיפוס העזר. מכניס נתונים לטיפוסי העזר מבלי לאתחל אותם. (ראו תכנית לדוגמה באתר)
5. אינו יודע כיצד לפענח פעולות ממשק. למשל, תלמידים זימנו פעולות מהממשק מבלי להעביר אליהן פרמטרים ככתוב בממשק, או לחילופין זימנו פונקציה כפרוצדורה ולהיפך. (ראו תכנית לדוגמה באתר)
6. ספר את כמות הטבעות הקטנות ואת כמות הטבעות הגדולות והכניס למוט המקורי רק סוג אחד מהם, למעשה את הסוג האחרון שהוצא. (ראו תכנית לדוגמה באתר)
7. ספר את כמות הטבעות הקטנות ואת כמות הטבעות הגדולות והכניס למוט המקורי את האותיות L ו- S שאינן טבעות. (ראו תכנית לדוגמה באתר)
8. התלמיד הגדיר טיפוס עזר מסוג "מוט" אך עבד עם מחשנית. (ראו תכנית לדוגמה באתר)

סיכום קשיים

- הבנה מערכתית של השאלה – למשל, תלמידים חשבו שמדובר כאן במגדלי הנוי. תלמידים לא עלו על הרעיון שניתן להשתמש בטיפוסי עזר ולהעביר אליהם את הטבעות ומשם להחזירן למוט המקורי.

המלצות כלליות להוראה בעקבות ממצאי המחקר

שאלה מספר 1

רצוי להדגים בפני התלמידים שאלות רבות ככל שניתן המכילות מורכבות של טיפוסים נתונים מופשטים מקוננים. אנחנו ממליצים לכלול שאלה מסוג זה בכל אחד מהמבחנים במהלך השנה כאשר רמת הקושי והפירוט עולה ממבחן אחד למשנהו. בכל פעם שנוסף טיפוס נתונים יש לבנות שאלה המכילה אותו והרכבה שלו בטיפוס אחר.

מומלץ לתרגל את השימוש במושגים "סביבת עבודה", "משק עברי", כתיבה של פעולות כולל התיאור וההנחות.

שאלה מספר 2

יש לתרגל את נושא העצים, לבנות תבניות לעבודה בעצים, כתיבת אלגוריתמים הקשורים לעצים – גם בסביבת העבודה. מומלץ לבנות תבניות קטנות שהתלמידים יוכלו להשתמש בהן באופן הדומה לפעולות הנמצאות בממשק העברי של הטיפוס עץ.

שאלה מספר 3

שאלה זו היא שאלה טיפוסית לטיפוסים נתונים מופשטים. מומלץ לתרגל עם התלמידים שימוש בממשק ובפעולות המצויות בו. ניתן להשתמש בדוגמאות המצויות בספר הלימוד, או לחילופין להמציא טיפוסים, לבנות להם ממשק ולהשתמש בו. רצוי לתרגל עם התלמידים במשך השנה כתיבה של תכניות בסביבת העבודה. התכניות הדרושות בדרך-כלל לטיפוסים נתונים מופשטים כלליים אינן קשות, מסתבר שהתלמידים פשוט אינם זוכרים.

סיבוכיות: יש לתרגל בכיתה את בניית פונקציית זמן ריצה וממנה לעבור לסיבוכיות או לחילופין הערכה של לולאות.

שאלה מספר 4

שאלה זו כוללת זימונים של אלגוריתם רקורסיבי. מילת המפתח כאן היא תרגול ושימוש בטבלאות/עצי מעקב.

באתר המרכז הארצי תוכלו למצוא גם פתרונות מומלצים לשאלות.

שאלה מספר 4

1. תלמידים הפסיקו לאחר הזימון ראשון.
 2. תלמידים התבלבלו בהצבות במערך לאחר ביצוע מציאת הגדול/הקטן ביותר במערך והחלפת המקומות.
 3. תלמידים אמרו שמוצא חציון.
 4. תלמידים שטעו בסעיף א' והפסיקו את ביצוע האלגוריתם לאחר השלב הראשון ענו כי האלגוריתם מוצא את האיבר הגדול ביותר והקטן ביותר.
- היו תלמידים שטענו כי האלגוריתם לא יפעל עבור $n < 7$ – האלגוריתם תמיד פועל, שואל שאלות ולפי התשובות מתבצע מיון או לא מתבצע מיון.

סיכום קשיים

- מורכבות הבעיה: בתוך האלגוריתם הראשי מופיעים קריאה רקורסיבית ושני זימונים לכל אלגוריתם משני. תלמידים התקשו להתמודד עם עומס זה.
- ריבוי הפרמטרים בזימון הרקורסיבי הוביל לבילבול אצל תלמידים.
- תלמידים הפסיקו לאחר זימון רקורסיבי אחד.
- תלמידים לא רושמים תוצאות ביניים ותוצאות סופיות – חלקם בודאי ניסה בדפי טיוטה משל עצמם (למרות האיסור על כך).
- תלמידים לא הבינו מה מבצע האלגוריתם מצא ולכן טעו בתרגיל כולו.
- חוסר יכולת – למשל, תלמידים חוזרים באופן מילולי על כל אחת מהפעולות של האלגוריתם הראשי. תלמידים לא מסוגלים להכליל.
- היו תלמידים שענו על סעיף זה נכון למרות שטעו בסעיף הקודם !!!!
- חוסר זהירות: תלמידים שהצליחו לגלות כי האלגוריתם ממין בסדר עולה את אברי המערך בין המקומות $p1$ ו- $p2$, ענו כי ימין בין המקומות 3 לבין 4 מקומות מהסוף מבלי להתייחס לערכו של 0.
- תלמידים לא הפרידו בין התחום שגדול מ- 7 או שווה לו לבין התחום הקטן מ- 7.
- תלמידים טענו כאשר K קטן מ- 7 האלגוריתם לא מתבצע – הטענה שגויה האלגוריתם מתבצע, אך עקב מענה שלילי על השאלה בתנאי העצירה לא מתבצע מיון כלשהו.

בניית משחק בפרדיגמה הפונקציונלית: התנסות בתהליך של פיתוח תוכנה

ד"ר תמר פז, המחלקה להוראת המדעים, הטכניון
אורנה מילר, אורט מעלות

1. מבוא

בניית פרויקט במסגרת יחידת הלימוד של תכנות פונקציונלי, אחת מהחלופות ליחידה השלישית [1], מאפשרת לתלמידים להתנסות בתהליך השלם של פיתוח תוכנה מורכבת בזמן קצר יחסית, ולהגיע לתוצר שהוא, בדרך-כלל, תוכנה של משחק. במהלך בניית הפרויקט חווים התלמידים את השלבים השונים של פיתוח תוכנה: ניתוח דרישות מערכת וזיהוי מרכיביה, פירוק המשימה לביצוע לתת-משימות ועידון הדרגתי של הפעולות המתבצעות במערכת, מימוש הפעולות בשפת התיכנות והרכבתן לביצוע התכנית השלמה. פיתוח פרויקט כולל גם בדיקות תקינות קלט ונכונות התכנית, תיעוד, ותיכון ממשק ידידותי למשתמש.

חשיפה מוקדמת זו לשלבי הפיתוח של תוכנה, מציגה בפני התלמידים תמונה כוללת ראשונה של התחום "עיצוב תכנה" (software design), ומושגים המוצגים מאוחר יותר לתלמידים (כמו מודולריות, שימוש חוזר ועבודת-צוות) הופכים משמעותיים יותר. התנסות בתהליך השלם וראיית המכלול חיוניים בטרם מרחיבים ומתעמקים בכל אחד ממרכיבי התהליך. ראיית התמונה השלמה של תהליך פיתוח תוכנה, אשר מזוהמת לתלמיד באמצעות התנסות חוויתית, היא הקדמה מתאימה, אם-כן, ליחידת הלימוד של עיצוב תוכנה.

2. שלבים בפיתוח תוכנה

ניתן לזהות בתהליך הפיתוח של פרויקט בתיכנות פונקציונלי את השלבים העיקריים בפיתוח תוכנה, תוך מתן תשומת לב לדרישות האופייניות מתוכנה. השלבים כוללים הגדרה של המשימה, הדרישות והאילוצים, פירוק המשימה לתת-משימות, ייצוג הנתונים במבני נתונים מתאימים, תיכון קשרי גומלין בין מרכיבי מערכת התוכנה, בנייתם, אינטגרציה שלהם, בדיקת נכונות, תיעוד ובניית ממשק ידידותי למשתמש.

עקרונות הפיתוח והדרישות המקובלות מן התוכנה כוללים מודולריות, שימוש חוזר, הכללה והפשטה ויכולת הרחבה. תהליך הפיתוח נעשה בדרך-כלל בעבודת צוות (של שניים או שלושה תלמידים).

שלבי הפיתוח קשורים זה בזה ואינם מתבצעים בהכרח באופן ליניארי. לעיתים קרובות, הדרישות ממערכת התוכנה משתנות ומתעדכנות, וההחלטות הנוגעות לשלב התיכון עוברות חשיבה מחודשת גם לאחר שהחל שלב המימוש. המערכת, לאחר השלמתה, עשויה לעבור שידרוג והרחבה (בהנדסת תוכנה מדברים על "מעגל פיתוח"). בתהליך פיתוח הפרויקט חווים התלמידים גם מצבים "אמיתיים" אלו.

בהמשך, נרחיב את הדיון בשלבים אותם עוברים מתכנתים בעת פיתוח תוכנה [3], ונדגים כיצד מתנסים בהם התלמידים במהלך עבודתם על הפרויקט בתכנות פונקציונלי (בסביבת התכנות DrScheme)¹.

1. סביבת DrScheme פותחה באוניברסיטת Rice כסביבה תכנותית פדגוגית ל-Scheme [5]. הסביבה מתעדכנת מדי מספר חודשים וניתנת להורדה בכתובת הבאה:
<http://www.cs.rice.edu/CS/PLT/packages/drscheme/download.html>

2.1 הגדרת דרישות

אפשר שפתרון של בעיה/תת-משימה ייבנה גם בכיוון ההפוך של תיכון מטה-מעלה (bottom-up design), תוך כדי הרכבה הדרגתית של פעולות בסיסיות, או פעולות מוכרות שנבנו בשלבים מוקדמים יותר של הלימוד.

פירוק לתת-משימות בשלב התיכון הוא הצעד הראשון לבנייה של תוכנה מודולרית. מודולריות מהווה עקרון מרכזי בהנדסת תוכנה, ויש לה השלכות משמעותיות על השלבים הבאים בפיתוח תוכנה. (בפרדיגמות שונות מודולריות עשויה להיות מושגת באמצעות גישות שונות).

מודול הוא חלק מתכנית, יחידה עצמאית במידה רבה ובעלת תפקיד אחד מוגדר, הפועלת על נתונים מסויימים ונושאת שם המשקף את המשימה שבאחריותה. מודול עשוי להיות מזומן על-ידי מודולים אחרים ולזמן מודולים נוספים.

מודולריות נתמכת ותומכת בתכונות מרכזיות נוספות המכוונות את העוסקים בעיצוב תוכנות: מודולריות מקלה על איתור שגיאות (debugging). כאשר תפקידו של כל מודול מוגדר וברור, קל יותר לאתר את המודול האחראי לבעיה נצפית. תלות קטנה ככל האפשר בין מודולים מאפשרת תיקון של טעות במודול אחד, ללא השפעת שרשרת על מודולים אחרים. מודולריות מאפשרת, בשלבים מתקדמים יותר של הפיתוח, לבצע בדיקת נכונות מסודרת וממוקדת. יעילות התחזוקה של תוכנה, דהיינו, הכנסת תיקונים ושינויים בהתאם לצרכים משתנים במהלך בנייתה ולאחריו, מושפעת ממידת המודולריות של התוכנה.

כאשר מספר אנשים (או קבוצות) שותפים בפיתוח תוכנה, מודולריות מאפשרת עבודה עצמאית של מתכנתים יחידים. חלוקת משימות בין אנשי הצוות ועבודה במקביל, לאחר שהוגדרו במדויק ממשקים וקשרי גומלין בין מודולים, היא התנסות שחווים התלמידים בעת הכנת הפרויקט בתכנות פונקציונלי.

שלב ראשון בפיתוח תוכנה הוא הגדרת הדרישות מן המערכת אשר מבקשים לפתח. הדרישות כוללות את התפקידים והמשימות אשר התוכנה אמורה למלא, והן תלויות בגורמים שונים כגון, סוג נתוני הקלט למערכת, אילוצי זמן פיתוח, עלויות חומרה ומהירות תגובה.

הדרישות צריכות להיות ברורות ושלמות ככל הניתן. בשלב זה הן צריכות לתאר את המשימות לביצוע ("מה") ולא את המימוש ("איך"). דרישות ברורות ומפורטות עוזרות גם בשלב של בדיקת תקינות התוכנה ועמידתה בציפיות.

בשלב הראשון בפיתוח הפרויקט בתכנות פונקציונלי בוחרים התלמידים בהתאם לנטייתם והעדפותיהם, את סוג התוכנה אשר ברצונם לפתח. אם בחרו בפיתוח משחק, עליהם להחליט בשלב זה מה יכלול המשחק שיפתחו: האם התוכנה תנהל משחק בין שני שחקנים אנושיים או בין שחקן אנושי למחשב? מה יהיו חוקי המשחק? האם מהלכי המחשב-כשחקן ייבחרו באופן אקראי או על-פי אסטרטגיה מסוימת? וכדומה.

הגדרת היקף המשימה ורמת מורכבותה מושפעים גם מדרישות בחינת הברורות ומאילוצי הזמן העומד לרשות התלמידים.

2.2 פירוק לתת משימות

אחת האסטרטגיות העיקריות להתמודדות עם מורכבות של בעיה היא חלוקתה לבעיות קטנות. פירוק הדרגתי של משימה למשימות קטנות יותר ויותר הוא חלק מתהליך של תיכון מעלה-מטה (top-down design) כשבסיומו מוגדרות פעולות אשר ניתן לממש באמצעות פונקציות בודדות.

בעת פיתוח הפרויקט, לאחר הגדרת הדרישות מן התוכנה השלמה, מנסים התלמידים לזהות במשימה הגדולה משימות חלקיות. בפיתוח משחק, למשל, מיפוי תת-משימות מתבצע באמצעות תיאור סכימטי של האלגוריתם למשחק. במקרה כזה, משימות חלקיות ראשוניות עשויות להיות: בניה של לוח משחק או של חפיסת קלפים, ערבוב קלפים וחלוקתם, בדיקה אם הושג ניצחון במשחק, וכדומה. כל משימה כזו עוברת פירוק נוסף, מעודן יותר.

2.3 ייצוג נתונים

שלב התיכנון כולל, בין השאר, בחירה של המבנים העיקריים בהם יאורגנו הנתונים. נבחנות צורות אפשריות שונות לייצוג, והשוואה ביניהן מובילה לבחירה של צורת הייצוג הנוחה ביותר ליישום הפעולות שהוגדרו.

התלמידים מחליטים בשלב זה כיצד לייצג כלי משחק שונים, כמו חבילות קלפים (רגילה, של טאקי או של משחק זיכרון), לוחות משחק (כיצד ייוצג לוח של "ארבע בשורה"?). האפשרות לייצג נתונים ביותר מדרך אחת, והשפעת אופן הייצוג על נוחות המימוש של פעולות שונות, הוא נושא הנידון בהרחבה ביחידת הלימוד של עיצוב תוכנה.

2.4 כתיבת הקוד בשפת תכנות

לאחר תכנון מפורט של המשימות אשר נדרש לממש, מגיע שלב הקידוד של המודולים שהוגדרו. בתכנות פונקציונלי מודול ממומש באמצעות כתיבת פונקציה או מספר פונקציות.

בשלב זה התלמידים יכולים לחלק ביניהם את עבודת הכתיבה ולחוות עבודת-צוות. זאת ועוד, מימוש מודולים שונים במקביל על-ידי תלמידים שונים ממחיש את היתרונות של חלוקת המשימה למודולים עצמאיים עם תלות מינימלית ביניהם.

בפיתוח הקוד, תלמידים עשויים לעשות שימוש בפונקציות שכתבו בשלבים מוקדמים יותר של הלימוד, בזמן שעסקו בבניית ארגזי כלים לטיפול במשימות שונות כמו, ערבוב ערכים ברשימה, עדכון רשימות ושימוש במספרים אקראיים. בשימוש בקוד שנכתב ונבדק בעבר, מתרגלים התלמידים עקרון חשוב בפיתוח תוכנה. שימוש חוזר חוסך בעבודה ובזמן פיתוח, ונתמך על-ידי מודולריות וכתיבת פונקציות כלליות.

2.5 הרכבה ושילוב המודולים

שלב ההרכבה דורש חיבור הדרגתי מטה-מעלה של המודולים השונים שפותחו. מודולים שונים האחראים לביצוע תת-משימות מקושרים ביניהם, עד לקבלת התוכנה השלמה.

בסביבה פונקציונלית, התכנית נבנית באמצעות הרכבה של פונקציות (ולא בזימון סדרתי של פעולות כמו בתכנות הפרוצדורלי) ועל-כן שלב זה מהווה אתגר משמעותי לתלמידים. תהליך ההרכבה עשוי לכלול, למשל, שילוב של צעדי שחקן אנושי וצעדי המחשב (אשר מומשו במודולים נפרדים) או שלבים שונים בקביעת המהלך הבא שיבצע המחשב. שלב ההרכבה דורש לעיתים קרובות כתיבה של פונקציות נוספות המתווכות בין המודולים.

כדי להקל בהתמודדות עם שלב ההרכבה, מתנסים התלמידים בשלבים קודמים של הלימוד בכתיבת פונקציות מתווכות, תוך פיתרון של בעיות מורכבות יותר ויותר הדורשות הרכבה של פונקציות באופנים שונים.

2.6 בדיקות תקינות

בדיקות נכונות ותקינות קלט מתבצעות בשלבי הפיתוח השונים ובמיוחד בשלבי ההרכבה והאינטגרציה. הרצה על סוגי קלט שונים כוללת טיפול ב"מקרי קצה" וערכים מיוחדים.

אחת מדרישות הפרוייקט היא טיפול במקרים של הכנסת קלט לא מתאים על-ידי המשתמש. למשל, הקלדת קלט תווי במקום קלט מספרי, או הכנסת ערכים שאינם בתחום הרצוי. הכתיבה הפונקציונלית מאפשרת בדיקה מיידית של כל פונקציה לאחר כתיבתה, ולכן בחינה של פעולת המודולים השונים נוחה לביצוע בכל שלב של הפיתוח.

2.7 תיעוד

תיעוד תמציתי, ברור ומדויק של התוכנה הוא מרכיב חשוב ביותר בפיתוחה. התיעוד הוא הדרך לאפשר למספר גורמים להתמצא במרכיבי התוכנה ומהווה אמצעי תקשורת ביניהם. כאשר הפיתוח מתנהל לאורך זמן, ומלווה בשינויים ותחלופה באנשים השותפים בפיתוח, חשיבותו של תיעוד מפורט ומעודכן הופכת קריטית ממש.

בתכנות פונקציונלי, העיסוק בתיעוד מלווה את כל שלבי הפיתוח והכתיבה של הקוד. כל פונקציה ומודול מלווה בתיאור תפקידו, הפרמטרים המועברים אליו, הערך המוחזר ממנו, ובמידת הצורך גם אופן מימושו וההנחות שנלקחו. תיק הפרויקט אשר מצרפים התלמידים לתכנית שפיתחו, מכיל גם הסברים הנוגעים לבחירת מבני נתונים, השיקולים בבחירתם, וכן הנחיות להפעלת התוכנה.

התלמידים נוכחים בחשיבות של התיעוד בפיתוח תוכנה גדולה, כאשר הם נדרשים לשלוט גם במודולים אשר פותחו על-ידי תלמידים אחרים, כחלק מן הדרישות בבחינת הברורות.

2.8 הרחבה

עיצוב של תוכנה רצוי שייקח בחשבון, כבר בשלב התיכנון, אפשרות של ביצוע שינויים עתידיים בה. אחד השינויים הוא הרחבה שלה למצבים כלליים יותר.

המושג של יכולת הרחבה של תוכנה, בא לידי ביטוי בתהליך הפיתוח של משחק כאשר ניתן לשנות את גודלו של לוח המשחק, מספר השחקנים המשתתפים, או האופן שבו מוגדר ניצחון במשחק. התלמידים לומדים כיצד בחירת ייצוג, הגדרת קבועים ומימוש פעולות באופנים שונים עשויים להשפיע על מידת הכלליות ויכולת ההרחבה של תכנית. במהלך ההתנסות הם רוכשים כלים להערכת איכות של תכנית, אשר אפשרות להרחבה היא אחד הקריטריונים החשובים בה.

3. הפרדיגמה הפונקציונלית כסביבה אידיאלית לפיתוח פרוייקט על-ידי תלמידים מתחילים

עד כאן דנו בשלבים בפיתוח תוכנה והדגמנו כיצד כל אחד מהשלבים בא לידי ביטוי בפרויקט שהתלמידים מפתחים במסגרת היחידה השלישית בתכנות פונקציונלי.

בפרק זה נדון בסיבות שהופכות את הסביבה הפונקציונלית למסגרת אידיאלית לפיתוח פרוייקט מורכב ורחב היקף, על-ידי תלמידים מתחילים, תוך פרק זמן קצר.

3.1. תחביר מינימלי

הפרדיגמה הפונקציונלית נשענת על מספר מצומצם של חוקים תחביריים. לכל הביטויים בסביבה יש מבנה אחיד (שם פונקציה ולאחריו פרמטרים אקטואליים), סוגריים מהווים את סימן הפיסוק היחיד והתוכנה מורכבת מאוסף של פונקציות בלבד (ללא תכנית ראשית). בנוסף, קיימים בשפה שני טיפוסים נתונים בסיסיים (אטום ורשימה) אשר די בהם כדי לתאר כל מבנה נתונים מורכב.

התחביר המינימלי והפשטות תורם לכך שניתן לפתח תוך פרק זמן קצר תכניות מורכבות, כגון משחקים ותכניות הנוגעות בתחום של אינטליגנציה מלאכותית. באופן מעשי, לאחר 12-15 שעות לימוד, התלמידים מכירים את פונקציות היסוד של סביבת העבודה, את טיפוסים הנתונים הבסיסיים בסביבה, ויודעים להגדיר פונקציות חדשות. ידע בסיסי זה מאפשר להם להתנסות בכתיבת משימות מורכבות הדורשות התמודדות עם כל המאפיינים של פיתוח תוכנה שנדונו בפרק הקודם (דוגמה למשימה עימה מתמודדים התלמידים בשלב זה תובא בסעיף 4.1).

3.2. הרכבת פונקציות כפעולה בסיסית

הרכבת פונקציות היא פעולה בה הערכים המוחזרים על-ידי פונקציה אחת או יותר משמשים כקלטות לפונקציה אחרת.

3.3. מבנה נתונים מורכב המאפשר גישה לכל איבר וגמישות במספר ובטיפוס האיברים

רשימת נתונים³ היא מבנה הנתונים המורכב היחיד המוצג ללומדי היחידה. באמצעות רשימה ניתן לתאר כל מבנה מורכב, בפרט מבני נתונים מופשטים (Abstract Data Types), ואף מבנים לא לינאריים כגון עצים. אחדות הייצוג של טיפוסים שונים מקלה על ביצוע פעולות עיבוד דומות עליהם (כגון חיפוש איבר). שלושה מאפיינים עיקריים של רשימה תורמים לכך שניתן לייצג באמצעותה מגוון של נתונים: גמישות במספר האיברים (כמו ברשימה מקושרת); שילוב של טיפוסים שונים (כמו ברשומה); גישה ישירה לכל איבר (כמו במערך).

א. גמישות במספר האיברים: מספר האיברים ברשימה אינו קבוע והוא יכול להשתנות בזימונים שונים של הפונקציה. גמישות זו במספר האיברים מקלה על פעולות התכנות ומשחררת את המתכנת מהצורך להצהיר מראש על מספר האיברים של הרשימה.

ב. שילוב של טיפוסים שונים: רשימת נתונים אינה מכילה בהכרח נתונים מטיפוס יחיד ואין צורך לקבוע מראש את טיפוס האיברים שהרשימה תכיל. אפשר שבזימונים שונים של הפונקציה, הפרמטר האקטואלי יהיה מטיפוסים שונים, ואפשר גם שהרשימה תכיל בו זמנית איברים מטיפוסים שונים, בכלל זה נכללות גם רשימות שחלקן או כל איבריהן הם עצמים רשימות.

ג. גישה ישירה לכל איבר: הסביבה הפונקציונלית מאפשרת גישה ישירה לכל איבר ברשימה לשם עידכון, העתקת ערכו וכיוצא בזה.

רשימה כטיפוס נתונים מורכב מאגדת בתוכה אם כך, את היתרונות של מערך עם אלו של רשומה ושל רשימה מקושרת. מעבר לכך, פעולות של הכנסת איבר לרשימה, בניית רשימה ואיחוד רשימות הן פעולות קלות לביצוע.

הרכבת פונקציות מאלצת את התלמידים לתכנת באופן מודולרי ולכתוב פונקציות ממוקדות אשר פותרות בעיות ספציפיות.

למשל, כאשר המשימה היא לכתוב פונקציה שמקבלת רשימת מספרים ומחזירה את מספר המספרים הגדולים מהממוצע, אסטרטגיה מקובלת תהיה לסרוק את רשימת הנתונים פעמיים: פעם אחת לשם חישוב הממוצע ופעם נוספת כדי למנות את מספר האיברים הגדולים מהממוצע.

הגישה המקובלת בפרדיגמה הפרוצדורלית לביצוע אסטרטגיה זו היא להציב בעת הסריקה הראשונה, במשתנה נוסף את הערך המקסימלי, ולאחר מכן בעת הסריקה השנייה, למנות את מספר האיברים הגדולים מהערך השמור במשתנה זה.

אולם אם נרצה לבצע את המשימה לעיל ללא שימוש בהוראות השמה, יהיה עלינו לחלק את המשימה לשלושה מודולים שימומשו באמצעות שלוש פונקציות: פונקציה אחת תחשב את הממוצע, פונקציה שנייה תבצע את הסריקה השנייה (תקבל את רשימת הנתונים ואת הממוצע ותמנה את מספר הגדולים מהממוצע) ופונקציה שלישית תתפקד כפונקציה ראשית: תבצע את ההרכבה ותפעיל את שתי האחרות.

למרות שסביבת העבודה מאפשרת ביצוע של הוראות השמה², החליטו מפתחי יחידת הלימוד [1] להסתיר יכולת זו מהתלמידים, ובכך להותיר את פעולות ההרכבה כאפשרות היחידה לפתרון בעיות בסביבה, ולחשוף את התלמידים לפרדיגמה תכנותית השונה באופן מהותי מזו שהכירו ביסודות [4]. כתוצאה מכך, הם מתנסים כבר מן השלבים הראשונים של לימוד הפרדיגמה בפירוק משימה לתת משימות ובתכנות מודולים עצמאיים לביצוע תת משימות.

3. בסביבת התכנות DrScheme רשימת נתונים מסומנת באמצעות הפונקציה quote (או בקיצור על-ידי גרש '). כלומר (arg1 arg2... argn). לדוגמה (76 88 90 100)

2. סביבת DrScheme מאפשרת השמה ישירה כמו בביטוי הבא (set! colors '(red blue white yellow))

3.4. סביבת עבודה אינטראקטיבית וידידותית

סביבת העבודה DrScheme מאפשרת להפעיל ביטוי בודד באופן ישיר ולא רק כחלק מתכנית שלמה. ביטוי זה יכול להיות מורכב מפונקציות יסוד או מפונקציות שהוגדרו על-ידי המשתמש. זימון מיידי של פונקציות מאפשר חקירה של פונקציות חדשות באמצעות בדיקה ישירה שלהן עם קלטים שונים, ומאפשר גם להפעיל בנפרד מודולים שונים המרכיבים את התכנית השלמה, מה שמקל על תהליך ניפוי השגיאות של התכנית.

להתמודדות המוצלחת עם מטלות מורכבות תורמת גם סביבת העבודה DrScheme שתומכת בכתיבה מודולרית גם באופן ויזואלי. הסביבה מספקת הודעות שגיאה ידידותיות ומובנות, ועוזרת להימנע משגיאות באמצעות התאמת סוגריים פותחים וסוגרים.

3.5. הפשטה ככלי באירגון ושליטה במורכבות

הפשטה היא כלי חשוב באירגון ושליטה במורכבות הן במתמטיקה והן במדעי-המחשב [6]. השליטה במורכבות מושגת באמצעות התעלמות זמנית מפרטים, ופיתוח "אבני בניין" שיהוו ממשק המאפשר בניית מערכות מורכבות [2].

הפשטה מהווה יצירת שפה ייחודית לצורך מציאת פתרון לבעיה ספציפית. תהליך ההפשטה חיוני כאשר כדי לענות על משימה נתונה אנו זקוקים ל"אבני בניין" שאינן חלק מהשפה.

במקרים אלה, פותר הבעיה בונה "מחסום הפשטה" שמחלק את תכנות המשימה לשתי משימות כמעט עצמאיות: מעל למחסום ומתחת למחסום. מעל למחסום הוא עוסק במציאת פתרון לבעיה המקורית במונחים של ההוראות שנבחרו על-ידו ("אבני הבניין" שבחר). מתחת למחסום, עליו לעסוק בפרטים של יישום ההוראות המופשטות הללו בשפת התכנות הנתונה [2,6].

כמובן, שבעיסוק בפרויקט תכנותי מורכב, יישום של הוראות מופשטות ברמה גבוהה עשוי לדרוש בנית הוראות מופשטות נוספות ומחסום הפשטה נוסף וכן הלאה, עד לרמת ההוראות היסוד המצויות בשפת התכנות.

הואיל והסביבה הפונקציונלית כפי שהוצגה ללומדים איננה כוללת הוראות השמה ולא כוללת משתנים (פרט לפרמטרים), היא מחייבת את הלומד, כבר משלבי ההיכרות הראשוניים, להגדיר מחסומי הפשטה ולפתח פונקציות שיהוו "אבני יסוד" בעת התמודדות עם פתרון בעיות מורכבות. זאת ועוד, השימוש הגמיש ברשימה לייצוג מגוון של טיפוסים נתונים מורכבים, ממנו נגזרות פעולות דומות על טיפוסים שונים, גם הוא כרוך בהפשטה.

התנסות זו בהגדרת מחסומי הפשטה מכינה את התלמידים להפשטה הנדרשת בעת פיתוח תוכנה מורכבת הן בשלב של פירוק המשימה לתת משימות הנפתרות באמצעות מודולים נפרדים, והן בשלב של הגדרת אופן ייצוג הנתונים.

3.6. רקורסיה כמבנה בקרה עיקרי

תרומה נוספת של הפרדיגמה הפונקציונלית ללומדיה קשורה בעיסוק האינטנסיבי ברקורסיה. הואיל ומבנה הבקרה העיקרי בפרדיגמה זו הוא רקורסיה (מבנה שמחליף את מבני החזרה בשפות אחרות), מתמודדים התלמידים כבר משלבים ראשוניים של היכרות עם הפרדיגמה עם כתיבת פונקציות רקורסיביות.

הפונקציות הרקורסיביות בהן עוסקים במסגרת היחידה של תכנות פונקציונלי הן בעלות מבנה אלגוריתמי אחיד (נפתחות בתנאי עצירה ולאחריו קריאה רקורסיבית אחת או יותר). מבנה אחיד זה מקל על תלמידים הנמצאים בשלבים הראשונים של שימוש ברקורסיה, לפתח פונקציות רקורסיביות ולעקוב אחר מהלך ביצוען.

משימה:

בלוח המשחק איקס-עיגול יש 3 שורות ו-3 עמודות.
כל משבצת מכילה אחד מבין שלושת הסימנים
הבאים: סימן של שחקן ראשון (x), סימן של שחקן
שני (o) וסימן של משבצת ריקה.
"לוח מנצח" במשחק הוא לוח בו בשלוש משבצות
רצופות (בשורה, בעמודה או באלכסון) מופיע הסימן x
או בשלוש משבצות רצופות מופיע הסימן o.
כתבו תוכנה שמקבלת לוח משחק איקס-עיגול
ומודיעה האם מדובר ב"לוח מנצח".

כבר בשלב מוקדם זה ניתן לדון בדרכים לפירוק משימה
לתת משימות תוך מתן דגש לכתיבת פונקציות כלליות.
למשל, בכדי לבדוק האם יש בלוח "שלושה מנצחת",
אפשר לכתוב 8 פונקציות (עבור 3 שורות, 3 עמודות
ושני אלכסונים). כל פונקציה כזו תקבל את הלוח
ותבדוק האם הערכים במשבצות המתאימות הם בעלי
סימן זהה. לחילופין, ניתן לכתוב שלוש פונקציות
מוכללות: אחת תטפל בשורות, פונקציה שניה תטפל
בעמודות ופונקציה שלישית תטפל באלכסונים.
אפשרות אחרת תהיה לכתוב פונקציה ברמת הכללה
גבוהה יותר אשר תקבל ערכים של שלוש משבצות (ולא
את הלוח השלם) ותודיע האם שלושתם שווים.

נושא נוסף שמעלה המשימה לעיל נוגע בהשוואה בין
אופנים שונים לייצוג נתונים, שכן יש צורך לייצג את
לוח המשחק (מטריצה) באמצעות רשימה. ניתן לבחור
למשל ברשימה בעלת תשעה איברים או ברשימה של
שלוש רשימות המייצגות את שלוש השורות בלוח.
בחירת הייצוג משפיעה על נוחות המימוש של הפעולות
שנבחרו (למשל, עדכון ערך של משבצת בלוח או בדיקה
אם משבצת נתונה היא ריקה). ההשוואה מתרגלת את
התלמידים בבחינת השיקולים אשר יש לקחת בחשבון
בעת בחירה של הייצוג המועדף.

בעקבות החשיפה לנושא הרקורסיה בשלב זה,
מתחילים התלמידים את לימודי יחידת "עיצוב תכנה"
כאשר המהות של רקורסיה מוכרת ומובנת להם, והם
יכולים להתמקד בפיתוח של פונקציות ופרוצדורות
רקורסיביות ובהפעלתן על מבני נתונים שונים, ויכולים
להתמקד גם באופנים שונים של שימוש ברקורסיה
(למשל, כשתנאי העצירה לא מופיע בתחילת הפונקציה,
או כשתנאי העצירה "חבוי")⁴.

כאן המקום להדגיש כי בפרדיגמה הפונקציונלית
רקורסיה איננה רק כלי תכנותי. רשימה שמהווה את
מבנה הנתונים המורכב היחיד בפרדיגמה היא מבנה
רקורסיבי שכן איבריה עשויים להיות רשימות שאיבריהן
עשויים אף הם להיות רשימות וכן הלאה. למעשה מדובר
בפרדיגמה שרקורסיה משמשת בה הן לתיאור מבני
נתונים (רשימות) והן לתיאור תהליכים (אלגוריתמים).

4. לימוד הדרגתי של פיתוח תוכנה – הלכה למעשה

הואיל ורכישת כללי התחביר של השפה תופסת חלק
קטן מן הזמן המוקדש ללימוד היחידה, עוסק החלק
הארי של הקורס בתכנות פונקציונלי בביצוע סדרה של
משימות תכנותיות שמורכבותן עולה בהדרגה. משימות
אלו מאפשרות לתלמידים להיחשף לשלבים ולעקרונות
שבפיתוח תוכנה, ולרכוש מיומנויות בתחום.

לאחר 12-15 שעות לימוד שבהן מכירים התלמידים
פונקציות יסוד, טיפוסים נתונים בסיסיים והגדרת
פונקציות חדשות, הם מסוגלים להתמודד עם כתיבת
משימות מורכבות כגון המשימה בעמודה הבאה.

4. הביטוי "תנאי עצירה חבוי" מתייחס לפונקציות
כדוגמת הפונקציה הבאה:

```
procedure f (n);  
begin  
  if n> 0 then begin  
    write ('*');  
    f (n-1);  
  end;  
end;
```

ההכרות עם שלושת הנושאים נועדה להקנות לתלמידים כלים לכתיבת תכניות מחשב רחבות ומורכבות יותר. פתרון בעיות הקשורות בכל אחד מן הנושאים, ובמיוחד בעיות המשלבות את שלושתם, מציב אתגרים לא פשוטים לתלמידים.

אף שהנושא לא נבדק מחקרית, התחושה הרווחת בין המורים המלמדים את היחידה של תכנות פונקציונלי היא כי היחידה תורמת למוכנות של התלמידים ללימוד עיצוב תכנה, במספר מישורים:

- חשיפה מוקדמת לתהליך הפיתוח של תכניות מורכבות.
 - בניית תוצר מוגמר (תוכנה שניתן לשחק בה).
 - התנסות משמעותית ביישום עקרונות של עיצוב תוכנה כמו מודולריות, תכנון מהכלל אל הפרט, ממשק ובדיקות נכונות.
 - כניסה "רכה" ללימוד נושא הרקורסיה.
- תרומה זו מאפשרת לגשר על הפער הניכר ברמת הקושי שבין יחידות הלימוד הראשונות לבין היחידה הרביעית, תוך כדי התנסות מושכת שתחושת הישג בצידה.

6. מקורות

1. לירון, לפידות, לוי ופז (1999). תכנות פונקציונלי – פרדיגמה תכנותית נוספת. ספר ראשון. טכניון.
2. Abelson, H., Sussman, G.J., & Sussman, J. (1996). *Structure and Interpretation of Computer Programs*. Second Edition. MIT Press, Cambridge, Massachusetts, London, England.
3. Bell, D., Morrey, I. & Pugh, J. (1992). *Software Engineering, a programming approach*. Prentice Hall.
4. Gal-Ezer, J., Beeri, C., Harel, D., and Yehudai, A. (1995). A High School Program in Computer Science. *IEEE Computer*. 28 (10), 73-80.
5. Felleisen, M., Findler, R.B., Flatt, M., & Krishnamuethi, S. (2001). *How to Design Programs. An Introduction to Computing and Programming*. Cambridge, Massachusetts: The MIT Press. available online: <http://www.htdp.org/2001-01-18/Book/>
6. Leron, U. (1987). Abstraction Barriers in Mathematics and Computer Science. In J. Hillel (Ed.). *Proceedings of the Third International Conference for Logo and Mathematics Education*. Montreal: Concordia University.

לאחר פיתוח תכניות עבור משימות כדוגמת המשימה לעיל לומדים התלמידים כתיבה של הוראות תנאי ופונקציות רקורסיביות רקורסיה. תוספת הוראות אלו מאפשרת להם לכתוב תכניות מורכבות יותר העונות על משימות מתקדמות, כמו אלגוריתמים למיון או ערבוב רשימת ערכים. כל משימה מהווה מיני פרויקט באמצעותו מתרגלים שלבים שונים בפיתוח תוכנה, ומוסיפה אלמנט חדש הקשור בפיתוח משחק. למשל, הדמיית הגרלה של מפעל הלוטו מתרגלת עבודה עם מספרים אקראיים. משחקים פשוטים כמו "התליין" ו"משחק הניחושים" מתרגלים כתיבת תכניות אינטראקטיביות ומשחק "מלחמה" בין שני שחקנים מתרגל רקורסיה הדדית והעברת התור בין שחקנים שונים.

לאחר ההכנה ההדרגתית למשימת הסיכום, מגיע השליש האחרון של הקורס שבו מתמודדים התלמידים עם פיתוח עצמאי של פרויקט מורכב הלקוח בדרך כלל מהתחום של אינטלגנציה מלאכותית. בפרויקט באים לידי ביטוי כל השלבים והעקרונות של פיתוח תוכנה שנילמדו במהלך הקורס. מרבית התלמידים בוחרים בפיתוח משחק, בדרך כלל מדובר במשחק קלפים כמו טאקי ופוקר או במשחק קופסה כמו איקס-עיגול, בול פגיעה, צוללות, רברסי, מנקלה, שולה המוקשים, מונופול, ארבע בשורה, לוטו, דומינו ועוד. תלמידים אחרים בוחרים לפתח תוכנה דמוית "אליזה" המדמה שיחה בין משתמש למומחה (רופא או פסיכולוג וכדומה).

המעבר המהיר לפיתוח תכניות מעניינות ושימושיות, מעודד תלמידים לתכנת בסביבה ולהפיק הנאה מרובה מפעילות זו. הנאה זו ניכרת בפרויקטים שהתלמידים מפתחים, ביצירתיות שהם מפגינים ובהיקף הרחב של התכניות. ניסיונו מראה כי פיתוח הפרויקט מהווה מסגרת מושכת ומאתגרת לתלמידים חזקים שבחרים להרחיב מעבר לדרישות הקורס ומשלבים בתכנותיהם גם פונקציות גרפיות ועבודה עם קבצים.

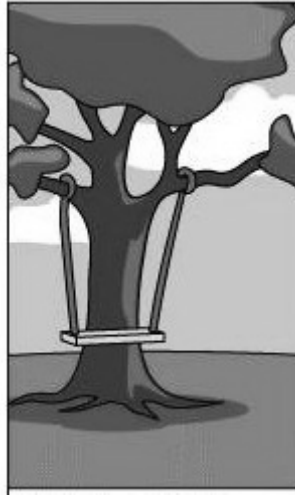
5. פיתוח פרויקט בתכנות פונקציונלי כהכנה ל"עיצוב תוכנה"

יחידת הלימוד "עיצוב תוכנה", בהמשך ל"יסודות מדעי-המחשב", משלבת לימוד של שלושה נושאים עיקריים: טיפוסים נתונים מופשטים, פיתרון בעיות אלגוריתמיות, ותיכון (design) של תכניות גדולות.

פיתוח תוכנה: נקודות מבט שונות



התכנון של מנתח המערכת



מה שהבין מנהל הפרויקט



ההסבר של הלקוח



הפעולות שהותקנו במערכת



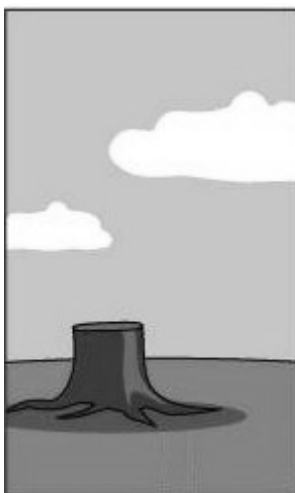
התעוד של הפרויקט



מה שכתב המתכנת



המוצר שהלקוח היה זקוק לו למעשה



התמיכה שניתנה לפרויקט



החשבון לתשלום שקיבל הלקוח

הוראת מדעי המחשב כאזוריים פוריים של מפגש תרבויות

ד"ר יפעת בן-דוד קוליקנט, האוניברסיטה העברית

פרופ' מוטי בן-ארי, מכון ויצמן למדע

תקציר^(*)

טורקל (1995) Turkle טענה שאפיסטמולוגית הידע והלמידה השתנתה, היות ואנו יכולים לחוות דברים דרך סימולציות מחשב, אשר אחרת היתה בלתי אפשרית, ואנו אף יכולים לטפח צורות קיום וירטואליות מרובות. היא קוראת לזה *תרבות הסימולציה* (culture of simulation), וטוענת שרוב משתמשי המחשב חיים *ברמת הממשק* (live on the interface). כלומר, הם מגבילים את פעילויותיהם לכאלו אשר כרוכות בהתעסקות עם ממשק התכנית ולא מתעמקים ביישום של התוכנה. המאפיינים של משתמשים אלו הינם שהם מציבים מטרות קצרות טווח ליצור תוצרים, לומדים על ידי ניסוי וטעייה דרך מניפולציות אובייקטים של הממשק והינם חסרי סבלנות להוראות והגדרות.

ברוסו (1997) Brousseau חקר את תרבות הכיתה של חינוך מתמטי. לפי ברוסו, מורים בונים *סביבות פדגוגיות*, להן הוא קרא גם *משחקים דידקטיים*. למידה משמעותית מתרחשת כאשר התלמיד/ה משחק/ת כדי לנצח, ולא רק כדי לשרוד על ידי עשיית המינימום ההכרחי כדי למלא את הדרישות המפורשות של המורה.

התנגשות תרבותית ואזוריים פוריים של מפגש תרבויות

התנגשות תרבותית בהוראת מדעי המחשב בצירוף התובנות של (ברוסו, בות' וטורקל), טענה הכותבת הראשונה שהוראה אפקטיבית דורשת שסביבת ההוראה בשלמותה, בעיקר המורה, תיחשב לגיטימית על ידי התלמידים (Ben-David Kolikant, 2004). היא תיארה את הוראת מדעי המחשב כמפגש בין וותיקים בשתי תרבויות אוריינות מחשבים שונות אשר מתארחות בתרבות שלישית, זו של בית הספר.

תיאוריות סוציו-תרבותיות על למידה מאפשרות לנו להסביר קשיי למידה בחינוך מדעי המחשב כתוצר של התנגשות בין תרבות המחשבים הלא פורמאלית של תלמידים לבין תרבויות מדעי המחשב האקדמאית והמקצועית. אנו מציעים את המונח **אזוריים פוריים של מפגש תרבויות** כדרך ניתוח והתגברות על אותם קשיי למידה. מטרת גישה פדגוגית זו היא לגשר על הפער בין תרבויות על ידי עיצוב התערבויות, אשר מוערכות על ידי חברי שתי התרבויות, ובד בבד מאפשרות ומעודדות את התלמיד להתנסות ולהעריך את הפרקטיקות של התרבות האחרת. הגישה הפדגוגית מודגמת על ידי ניתוח מפורט של אפיזודות, הלקוחות מהקורס "חישוב מקבילי ומבוזר", שם אנו מראים שאזוריים פוריים של מפגש תרבויות יכולים להיות מוצלחים לשיפור משמעותי של הלמידה.

תיאוריות סוציו-תרבותיות בהוראת מדעי המחשב

תרבות הוגדרה על ידי האנתרופולוג קליפורד גירץ (Clifford Geertz) כרשת של משמעות וכקבוצה של תוצרים (artifacts) (מצוטט ב Booth (2001, p.3)). Booth (2001) הגדירה שלוש תרבויות תכנות:

- התרבות האקדמאית (academic) של מודלים מופשטים של חישוב.
- התרבות המקצועית (professional) שחבריה מעצבים ומיישמים מערכות.
- התרבות הלא פורמאלית (informal), אשר כוללת הן חובבנים (כמו האקרים) והן ציבור כללי של משתמשי מחשב.

(*) המאמר עובד לפרסום בידי שלי גוטנר על בסיס מאמר נרחב יותר של המחברים.

לכן, מחנכי מדעי המחשב עומדים בפני דילמה: כיצד הם יכולים לזכות בלגיטימציה של תלמידים מהתרבות הלא פורמאלית מבלי לאבד את התרבות האקדמאית, אשר ממנה הם מגיעים? במילים אחרות, כיצד הם יכולים לשכנע את התלמידים לחצות את הגבולות התרבותיים ולקחת צעד לקראת התרבות האקדמאית של מדעי המחשב, אימוץ והשגת יעדים על ידי שימוש בכלים אשר אופייניים לתרבות זו?

אזורים פוריים של מפגשי תרבויות

אנו טוענים שפתרון להתנגשות התרבויות נעוץ בעיצוב זהיר של הוראה כאזורים פוריים של מפגשים תרבותיים (אפמ"ת). אזורים פוריים של מפגשי תרבות הינם מסגרת פדגוגית שבה פעילויות הוראה מעוצבות להפוך את תרבות היעד (אקדמאית ו/או מקצועית) אטרקטיבית יותר לתלמידים ולספק לתלמידים הזדמנויות להתחייב לפרקטיקה של תרבות היעד. (לשם קיצור נשתמש במושג אפמ"ת גם לפעילות אשר מעוצבת בתוך מסגרת זו). הכוונה של פעילויות אלו היא לעודד ולאפשר לתלמידים לחצות את הגבול התרבותי לעבר תרבות היעד. על מנת לעצב אזורים פוריים של מפגשי תרבויות, מורים חייבים להכיר את שתי התרבויות השייכות לתחום הנושא הספציפי, כך שיוכלו לראות את תוכן הקורס כפי שהתלמידים יראו אותו, ולא רק באופן שמורה רואה אותו. נקודת השקפה דואלית זו הכרחית במטרה לעצב פעילויות דידקטיות אשר מעודדות, מעוררות מוטיבציה ומקלות על התלמידים לחצות את הגבולות התרבותיים.

מתודולוגיה

פיתחנו קורס בחישוב מקבילי ומבוזר עבור תלמידי תיכון בכיתות יב' אשר לומדים תכנית מתקדמת של מדעי המחשב (5 יח"ל). קורס זה עוסק במערכות תוכנה מרובות ישויות. כל ישות מורכבת מרצף הוראות אשר מבוצעות אחת אחר השניה, אך, כל הישויות מבוצעות במקביל וגורמות לאי-דטרמיניזם, מכיוון שישנם תסריטים רבים של ביצועי המערכת.

המורה הוא נציג של תרבות מדעי המחשב האקדמאית-מקצועית, בעוד שהתלמיד/ה מחובר/ת עמוקות בתרבות התכנות הלא פורמאלית. במדעים אחרים, תמצאו תלמידים ספורים בלבד עם ניסיון נרחב בתרבות הלא-פורמאלית של אותו מדע. למשל, כילדים יתכן כי שיחקו עם ערכות כימיה, אך רק מעטים מבינו החשיבו עצמם כותיקים בתרבות הלא פורמאלית של הכימיה, הדרך שבה היום תלמידים חשים לגבי התרבות הלא פורמאלית של מדעי המחשב. כוונתיים בתרבות הלא פורמאלית, תלמידים מאמינים כי יש להם את הזכות לשפוט את הלגיטימיות של כל פעילות, כמו גם את תכנית הלימודים כולה. כתוצאה מכך, הכותבת הראשונה טענה שמורי מדעי המחשב מתמודדים עם התנגשות תרבויות וקשה להם לבסס את הלגיטימיות שלהם.

מה שנחשב כחשוב בתרבות הלא פורמאלית זה השימושיות המיידית של תוצרי מחשב; התוצאה היא שהמושגים המרכזיים של התרבות האקדמאית – אלגוריתמים, מודלים חישוביים, ומבנים לוגיים של מערכות חומרה ותוכנה – חסרים לגיטימציה. באופן דומה, חברים בתרבות הלא פורמאלית, אשר חיים בממשק של טורקל, הינם בעלי הרגלי עבודה השונים באופן עקרוני מהרגלי העבודה האופייניים של התרבות האקדמאית של מדעי המחשב. בהסתמכם על ניסיונם האינטנסיבי עם טכנולוגיה, הם יידעו שפתרון בעיות של מחשוב נעשה בהצלחה על ידי מניפולציה טכנולוגית מחשבים דרך ניסוי וטעייה. לעומת זאת, הרגלי העבודה אשר מקושרים עם התרבות האקדמאית והתרבות המקצועית כוללים ניתוח ועיצוב לפני קידוד, חשיבה ביקורתית על המשימה כולה, חוסר סובלנות לטעויות, בחירה של המודל החישובי הראוי למשימה ותייעוד לשם תחזוקה. פעילויות אלו נתפסות כמוזרות ולא רלוונטיות לאינטרסים של התלמידים בעולם המחשבים.

אנו משערים שחוסר לגיטימציה זה, הנוצר מההתנגשות התרבותית, יכול להוביל את התלמידים לסרב להשתתף בפעילויות למידה, או לכל היותר להשתתף רק בהיקף הנדרש כדי לשרוד את הסביבה של המשחקים הדידקטיים.

אי לגיטימיות של משימות

בתוך התרבות האקדמאית, בעיות לרוב מנוסחות כסיפור דמיוני; אחת המוכרות ביותר היא *בעיית הסועדים הפילוסופיים* (ראו נספח). בעיות מסוג זה הן מופשטות. כלומר, הן אינן בעיות קונקרטיות, אלא בעיות אשר הסיטואציות שלהן מדגימות עקרונות וטכניקות הישימים למערכות אמיתיות. כאשר פיתחנו את הקורס שלנו, החלטנו ללכת בעקבות הפרקטיקה של הקהילה האקדמאית ולהשתמש בבעיות דמיוניות, כגון בעיית הפילוסופים הסועדים. אנו קיוונו כי השפה הדמיונית תגדיל את המוטיבציה ותספק אותנטיות לתרבות האקדמאית.

הופתענו לגלות קשיים עם בעיות אלו. הקשיים נגרמו משתי סיבות עיקריות: ראשית, רוב התלמידים לא פירשו את הבעיה בתוך המודלים החישוביים אשר לימדנו אותם, ובמקום זאת העדיפו להישאר בתוך התחום של הסיפור. תלמידים רבים פתרו בעיות על ידי הנחת השערות בלתי ריאליות או על ידי שינוי הבעיה לטריטוריה אחרת. התלמידים צולמו בוידיאו במהלך עבודתם והנה מספר תשובות טיפוסיות: (א) י. הציע שהפילוסופים ישתמשו בעיניהם כדי להחליט אם המזלג פנוי; (ב) האלגוריתם של ג' הנחה את הפילוסופים להתנהג באופן תרבותי; (ג) י. ו-א. הציעו לחסום פילוסוף אחד לנצח, להוסיף מזלג או לאפשר לאחד מהפילוסופים להשתמש בידיו. באופן ברור, תשובות אלו הן משמעותיות רק בתוך הסיפור ולא בתוך מודל חישובי הגיוני כלשהו! לעומת זאת, גישה של ותיק לבעיית הפילוסופים הסועדים תכלול טרנספורמציה של הבעיה למודל חישובי פורמאלי.

שנית, תלמידים רבים הביעו באופן מפורש ספקות לגבי החשיבות של בעיה מסוג כזה. התלמיד א.פ. יזם שיחה, "לטובת עתיד התלמידים", ואמר:

אני לא מבין מה הטעם בלימוד דברים אלו, כמו המודל החישובי, דרישות, הוכחה וכל הדברים האלה, כל הפילוסופים הבעייתיים האלו. מה הטעם? מתי אתה כותב משהו? תכנית? בקורסים אחרים אתה כותב, אתה עושה דברים.

יתכן כי עיצוב המערכת יצריך תיאום בין ישויות, למשל, כאשר ישויות אלו חולקות משאב. תיאום מושג על ידי הוספת הוראות לישויות באופן המבטיח שהישויות לא יוכלו להשיג גישה לאותו משאב בו-זמנית. הנושא של מקביליות מספק *הזדמנות ייחודית* להדגים את הכוח של התרבות האקדמאית והתרבות המקצועית, מכיוון שהפרקטיקות של התרבות הלא פורמאלית, כמו ניסוי וטעייה, הן חסרות תועלת לחלוטין בפיתוח ואימות של פתרונות אלגוריתמים מקבילים ותכניות מקבילות.

ביצענו מחקר במטרה לגלות קשיי למידה אשר מתרחשים במהלך הקורס. החקירה בוצעה במהלך שנתיים עוקבות שבהן הקורס נלמד. בשנה הראשונה, תיעדנו את קשיי הלמידה: צפינו בשיעורים של כיתה אחת לאורך כל התקופה של הקורס, אספנו מבחנים, צילמנו בוידיאו תלמידים במשימות מעבדה של פתרון בעיות, וראיינו תלמידים. ניתוח הנתונים הוביל לעיצוב מחדש משמעותי של הקורס. הגרסה המשופרת נלמדה בשנה השניה, אשר במהלכה חזרנו על איסוף הנתונים וניתוח במטרה לוודא את שיפור הלמידה.

קודם ללמידת הקורס, התלמידים כבר סיימו קורס דו-שנתי בנושא יסודות מדעי המחשב וקורס שנתי בפרדיגמת חישוב נוספת. כלומר, לתלמידים היה ניסיון משמעותי עם תכנות והכירו כבר מושגים בסיסיים במדעי המחשב, כמו תכנית, בעיה אלגוריתמית, ואלגוריתם. כמו כן הם למדו שתי שפות תכנות ולכן המושגים הבסיסיים הקשורים בעיצוב ויישום תכניות היו מוכרים להם היטב.

התנגשות תרבותית כמסגרת לניתוח קשיי למידה

כעת נתאר סיטואציות, אשר אנו טוענים כי הן חולקות את אותו המאפיין: כאשר ניצבים בפני משימות תכנות או סיטואציות אחרות הכרוכות במחשב, תלמידים ניגשים לסיטואציה ומייצרים תוצאות אשר אינן לגיטימיות בעיני המורה (או בעינינו כמדעני המחשב וכמחנכים). מנגד, התלמידים עצמם לא נותנים לגיטימציה למטרות אשר אותן היו אמורים להשיג, ולא לאמצעים אשר סופקו להם כדי להשיג מטרות אלו. לכן, הסקנו שסיטואציות אלו גרמו להתנגשות תרבותית.

במקום רק לבקש פתרונות, נתנו לתלמידים חמישה פתרונות אפשריים וביקשנו מהם להעריך את הנכונות של כל אחד מהם, דבר המאפשר לנו לבחון טווח רחב של אי-הבנות אפשריות. פתרון אפשרי אחד מוצג בתרשים 1. פתרון זה מבטיח כי Op1 תתבצע לפני Op2 ו- Op3, אבל הוא גם מאלץ את Op2 להתבצע לפני Op3. למרבה ההפתעה, 44% מהתלמידים ציינו שפתרון זה הוא נכון. באופן דומה, 53% מהתלמידים קיבלו פתרון אחר בו כל ההוראות הישות E1, לא רק Op1, בוצעו לפני כל ההוראות של שתי הישויות האחרות (תרשים 1).

נזכיר כי בהצגת הבעיה נאמר לתלמידים במפורש שהפתרון הוא נכון רק אם הוא מאפשר את כל התסריטים, המספקים את דרישת התיאום. בנוסף לכך, ניתן לתלמידים מספיק זמן לעבוד על הבעיה, ולא נראה כי הם לא הבינו את ההוראות. לכן, זה מפתיע כי 44% ו- 53% מהתלמידים החליטו שהפתרונות הלא נכונים היו נכונים, ואנו לא יכולים להניח כי היה זה חוסר ביכולת קוגניטיבית או חוסר מאמץ אשר גרמו לבעיה. בנוסף לכך, ניתוח ההסברים שלהם חושף שהתלמידים היו מודעים לאילוצים הלא הכרחיים, ועדיין שפטו את הפתרונות כנכונים ברגע שהבחינו שהדרישה הראשונה התקיימה.

לכן, הסקנו שזה היה מקרה של התנגשות תרבותית. במודל המקבילי אשר הם למדו – מודל הנמצא בשימוש התרבות האקדמאית והתרבות המקצועית – פתרון נכון הוא פתרון בו רק תסריטים ספציפיים מסוימים (אלה שהוגדרו כתסריטים רעים על-ידי דרישות התיאום) לא יתרחשו, בעוד שהוא מאפשר לכל התסריטים שאינם אסורים להתרחש. באופן ברור, דרישה זו, אשר בדרך כלל מרומזת בספרי הלימוד (או לפחות אינה מודגשת) ואשר אותה כתבנו באופן מפורש, לא מוכרת לחברי התרבות הלא פורמאלית, וגם לא נשפטה כחשובה. גישתם נראתה כזאת, שכל עוד לא קורה משהו רע (על המסך), התכנית היא טובה; הם לא מרגישים שהם אחראיים להבטיח שכל התסריטים הטובים (לא רעים) יכולים להתרחש.

הסבר הגיוני לחוסר ההערכה של התלמידים לבעיות אלו הוא שהתלמידים חיים "על" הממשק (living on the interface) ומחפשים ליצור תוצרים שימושיים. הם רואים את הסיפורים האלו כראיה לכך שהמורה חיה בעולם האגדות ולוקחים זאת כהוכחה לחוסר הרלבנטיות של לימודי מדעי המחשב האקדמאיים!

במקרה זה ניתן להבחין בין שתי הגדרות אסתטיקה שונות (Turkle, 1995): ההגדרה של תרבות מדעי המחשב האקדמאית אשר רואה בבעיות כמו הפילוסופים הסועדים כיפיות, וההגדרה של התלמידים מהתרבות הלא פורמאלית, אשר אינם רואים יופי כלשהו בבעיות אלו, אלא בתוצרים טכנולוגיים הידיוניים למשתמש. העדפה זו מתורגמת להתנהגות מעשית מצידם של התלמידים, כמו ההחלטות אם להשתתף באופן מלא, לשרוד או להישאר בשוליים. הסקנו שבפני התלמידים צריכות להיות מוצבות מטרות, המוערכות על ידיהם, ושבהלך ההוראה עלינו ללמד אותם בהדרגה את היופי ואת הכח של התרבות האקדמאית, כך שבסופו של דבר הם יעניקו לגיטימציה למטרות טיפוסיות של תרבות היעד.

שתי הגדרות של נכונות

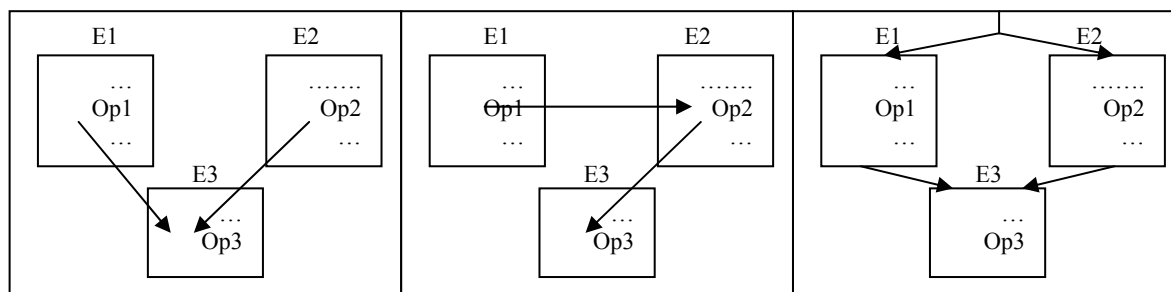
הסיטואציה המתוארת בחלק זה מדגימה את המשמעות השונה, שחברים מכל אחת מתרבויות אוריינות המחשבים ייחסו לאותה המילה, נכונות, כפי שהשתקפה בפרקטיקה שלהם. ממצא זה עקבי לטענה של ויגוצקי, שלמידה דורשת כלים תרבותיים – בעיקר שפה – עם קשריהם למוסדות חברתיים (Kozulin, 1998).

הבעיה הבאה הוצגה בפני 138 תלמידי מדעי המחשב (הפתרון הנכון מוצג באופן סכמתי בתרשים 1א, בהמשך). מערכת מורכבת משלוש ישויות E1, E2 ו- E3, המתבצעות במקביל. רצף ההוראות של הישויות כולל את ההוראות Op1, Op2 ו- Op3, בהתאמה. הוסיפו הוראות לתכנית במטרה להבטיח כי (א) Op1 תתבצע לפני Op2 ו- Op3, אבל (ב) אין להוסיף אילוצים אחרים. כלומר, פתרון הוא נכון רק במידה והפתרון מאפשר את כל התסריטים המספקים את דרישת התיאום ורק תסריטים אלו. הסבירו את התכנון שלכם.

כלומר, הדרישה לעבוד בתוך מודל החישוב הפורמאלי. מכל מקום ההטמעה של ההגדרה האקדמאית של מדעי המחשב של נכונות על ידי התלמידים היא מרכיב קריטי בהוראת מדעי המחשב, אשר קשור להרגלי עבודה שתלמידי מדעי המחשב מאמצים בהדרגה, בו זמנית עם ערכים אותם הם לומדים להעריך. נקודת המבט התרבותית שולטת בשיפוטם של התלמידים לגבי מה נכון ומה שגוי, מה ראוי שישקיעו מזמנם ללמוד ומה "נחמד שיש" אך לא הכרחי בפרקטיקת התכנות.

אכן, בדיון אשר נערך בכיתה, התלמידים נתנו טיעון זה בדיוק בהתייחסם לתנאי השני (העדר של אילוצים לא ספציפיים) או כתכונה חביבה אבל בהחלט לא קריטית, של הפתרון כקפריזה קטנונית מצידה של המורה! שוב, נראה כי התלמידים לא נתנו לגיטימציה למטרות שהצבנו בפניהם. הסקנו כי צפינו בשתי הגדרות שונות של נכונות, המבוססות על נקודות המבט התרבותיות של התלמידים מול זו של המורה.

בראיה לאחר, היה נאיבי מצידנו לצפות שתלמידים יטמיעו את ההיבט הזה של התרבות האקדמאית,



איור 1א. דרישת התיאום הנכונה. איורים 1ב ו-1ג דרישות עם אילוצים מיותרים

אזורים פוריים של מפגש תרבויות כמסגרת לשיפור פדגוגי במדעי המחשב

עיצוב משימות

היות והתלמידים שופטים פעילויות מנקודת מבטם התרבותית, הרעיון היה להציב מטרות אשר יראו בעיניהם לגיטימיות. מנגד, המשימות עוצבו בדרך כזו שנדרש מהתלמידים להתעסק בפרקטיקת היעד של העולם האקדמאי של מדעי המחשב במטרה לפתור אותן בהצלחה, מה שגרם לכך שיעריכו את פרקטיקת היעד.

לתלמידים ניתנה תכנית מקבילית אשר מיישמת משחק פשוט, שבו צריכים לשגר טיל כדי ליירט חייזר; הם התבקשו להרחיב את התכנית שתאפשר לשגר שני טילים.

תחילה נתאר בפירוט משימה אחת, ונסביר את השיקולים שעשינו כדי לרכוש את הלגיטימציה של התלמידים, בתקווה שבפתרון המשימה הם יתעסקו בפתרון על ידי שימוש בפרקטיקות של תרבות היעד.

לאחר מכן, נסביר כיצד תכננו את רצף המשימות במטרה לזכות בהדרגה בהערכה של התלמידים הן ליופי והן לכוח של הפרקטיקה.

לאחר מכן נסביר את עיצוב סביבת ההוראה ונדון בתפקיד המורה באזורים פוריים של מפגש תרבויות כמתווך בין שתי התרבויות כמסייע במעבר לתרבות היעד.

האזור של מפגש התרבויות היה הצגת הבעיה בצורה שהיתה מוכרת לתלמידים ואשר תאמה לציפיות שלהם מבעיות בעולם המחשבים. הכח של ההפשטה הוצג בהדרגה ורק אחרי כן הצגנו לתלמידים את האופן המופשט בו מוצגות בעיות בעולם האקדמי.

אספקת בסיס למשא ומתן על הפרקטיקה

חיפשנו דרך להקל את המעבר מהפרקטיקה של התלמידים לפרקטיקה האקדמאית. לשם כך יצרנו סיטואציות חינוכיות אשר מקלות על המשא ומתן על משמעות בהיבטים הבאים של הפרקטיקה: (א) משא ומתן על המשמעות של מילים אשר קשורות לפרקטיקה, כמו ההגדרה של נכונות, ו-(ב) משא ומתן על ההשפעה המעשית של ההבנה שלהם והתוצרים שלהם במטרה לעודד את התלמידים להעריך מחדש את האמונות שלהם ואת אסטרטגיות פתרון הבעיות.

הבעיה הבאה הוגשה כבסיס למשא ומתן על המשמעות של נכונות וההסתעפויות המעשיות שלה. שימו לב שהבעיה זהה לבעיה שבה השתמשנו כדי לתאר את שתי ההגדרות של נכונות, אך ניסחנו אותה מחדש, כך שתהיה שייכת לעולם המחשבים המוכר:

למהנדס תוכנה ניתנה מטלת יישום של אלגוריתם של תיאום, אשר מעניקה עדיפות גבוהה יותר לתכניות שהוגשו על ידי חברי פקולטה מאשר שנתנה לתכניות אשר הוגשו על ידי תלמידי תואר ראשון או תלמידי תואר שני. אלגוריתם שתכנן מעניק עדיפות גבוהה לתכניות מחברי הפקולטה, עדיפות בינונית לתכניות של תלמידי תואר שני ועדיפות נמוכה לתכניות של תלמידי תואר ראשון. לאחר שתלמידי התואר הראשון התלוננו, תבע ראש המחלקה מהמהנדס לשנות את התכנית, כך שתלמידי התואר הראשון ותלמידי התואר השני יקבלו את אותה העדיפות; מנגד, המהנדס טען שהוא מילא נאמנה את דרישות התוכנה. מי צודק?

בתכנית המקורית, דרישות התיאום מולאו בצורה נכונה. שימו לב שהתלמידים התבקשו לייצר מוצר תוכנה שהיה מוצר מוכר, משחק מחשב. כתוצאה מכך, שיערנו שהם ברצון יעסקו בבעיה זו, מכיוון שעבודה עם משחקים, עם הרבה ישויות מהווה עניין מידי עבורם.

בנוסף שיערנו שהתלמידים, הרגילים לחיות ברמת הממשק, פשוט יגזרו וידביקו (cut-and-paste) את הקוד לטיל אחד כדי ליישם את פעולות הטיל השני. מכל מקום, הפעילות עוצבה כך שגישה נאיבית זו לא תעבוד, התלמידים ייווכחו בצורך בניסוח דרישות תיאום פורמאליות וביישום של פתרון בתוך המודל הפורמאלי. אזור המפגש הוא העולם התרבותי המוכר של משחק מחשב המתנגש בדרישות וביישום של התרבות האקדמאית.

שתי ההשערות נמצאו נכונות: רוב התלמידים בחרו מרצונם לבלות את זמנם החופשי עד אשר פתרו בהצלחה את המשימה, אבל הם היו צריכים "לחצות את הגשר" ולהשתמש בשיטות פורמאליות כדי להשיג הצלחה זו. אנו רואים זאת כאזור פורה של מפגש תרבויות היות וזה גרם לתלמידים לחתור ליכולת בתרבות היעד האקדמאית.

כתוצאה מהניסוי שלנו, הצבנו לפנינו את המטרה לספק גשר מההתעסקות עם תוצרים ברמת הממשק לבעיות אקדמאיות מופשטות. למטרה זו, דחינו את הצגת בעיית הפילוסופים הסועדים לטרימסטר האחרון של הקורס. כמו כן, הוצאנו מבעיות אחרות, כמו בעיית הקטע הקריטי (critical section), מההפשטה הלא ריאלית שלהן ומהשפה הדמיונית (עוגות וסכינים, ארנבים וגזרים) בהן השתמשנו בגרסה הראשונית של הקורס. במקום זאת, חיפשנו בעיות ריאליות הכוללות מקבילות ותאום, כגון כספומטים, משרדי כרטיסים, חשבונות בנק וכך הלאה. בנוסף לכך, רק אחרי שכל בעיה הוצגה ונפתרה בתוך מודל חישוב פורמאלי, הכללנו בעיה מופשטת מתוך זה ודנו בפתרון מופשט. מאוחר יותר, כאשר מודל חישובי נוסף הוצג, פתרנו את כל הבעיות שוב, ועל ידי כך חיזקנו את שימושיות והמשמעות של הבעיה המופשטת.

מורה: האם בדקת את זה?

תלמיד: כן.

מורה: איך?

תלמיד: הרצתי את זה.

מורה: ו...?

תלמיד: זה עובד. זה מדפיס מעט זבל בחלק העליון

של המסך אבל זה לא חשוב.

מורה: [ניגשת למחשב של התלמיד] תראה לי.

תלמיד: [מריץ את התכנית ומכניס קלט; מצביע על

המסך] הנה, זה עובד.

מורה: האם זה הפלט לו ציפית?

תלמיד: אני לא יודע

מורה: זה לא מקצועי; תכנן את הקלט ואז תבדוק

אם הפלט תואם לציפיות שלך.

תלמיד: [מחשב עם עפרון ונייר; מריץ ומכניס קלט;

מצביע על המסך] אההה!... לא, זה לא

[הפלט לא היה מה שהוא ציפה]

מורה: אוקי, תתקן את זה בבקשה.

תצפית הבטיחה לנו שתלמיד זה השקיע מאמץ משמעותי בניסיונות שלו לפתור את הבעיה לפני שמסר את התדפיס למורה, כך שהדרך היחידה אשר יכולה להסביר מדוע הוא אפילו לא טרח לוודא את הנכונות של הפלט הינה שהוא היה מבוצר עמוקות בתרבות הלא פורמאלית, אשר אינה מעודדת פעילויות שכאלה.

אנו טוענים שהמורה יצרה בזירות גשר לתלמיד, עודדה אותו להעריך את השיטות שלו והציעה לו התנהגות חלופית. ראשית, המורה לא אמרה לו שהוא טעה, ולא הגיבה ישירות להצהרת התלמיד: "זה עובד, זה מדפיס מעט זבל... אבל זה לא חשוב", למרות שהיא ידעה שהזבל היה תוצאה של הטעות בתכנית של התלמיד. במקום זאת, האינטראקציה כללה שאלות אשר עודדו את התלמיד לחשוב על השיטה שלו לבדיקת תכניות. לבסוף התלמיד נוכח שהשיטה שלו היתה לקויה. תלמיד זה ניגש מאוחר יותר לשולחן המורה והסביר - מבלי שהתבקש לכך - מה היה המקור של הזבל, והפעם השתמש במינוח מקצועי.

ברגע שהבעיה הוצגה כבעיה הלקוחה מהעולם המחשבים האמיתי, התלמידים יכלו להזדהות עם הטענות של שני הצדדים. מהנדס התוכנה (כמו 44% ו- 53% מהתלמידים) ניגש לבעיה מגישת התרבות הלא פורמאלית, אשר אנו יכולים לכנות כ"מקווים לטוב": הוא סיפק פתרון מינימלי אשר היה תקף במקרה הטוב. ראש המחלקה ציפה מהמהנדס לגשת לבעיה מגישת תרבות מקצועית, אשר אנו יכולים לכנות כ"מתכווננים לגרוע מכל": הפתרון אמור לעבוד בכל המקרים ואילו ציפים אינם יכולים להתווסף באופן שרירותי. בכלל, הרגלי המחשבה בתרבות האקדמאית והתרבות המקצועית מובילים לדגש על ניתוח ועיצוב, היות ולא ניתן להסתמך על ניסוי וטעייה אם מתכוונים "להתכונן לגרוע מכל".

אזור פורה של מפגש תרבויות ישכנע את התלמידים לשנות את הרגלי החשיבה והעבודה שלהם, משימה לא קלה כאשר הרגלים אלו הוכחו כמעשיים ואפקטיביים בפעילויות העבר שלהם בתוך תרבות המחשב הלא פורמאלית וגם תרבות בית הספר.

אנו טוענים שהמפגש היה מורה מכיוון הוא אפשר לנו להסביר את ההגיון והשימושיות המונחות בבסיס הדרישה "אילו ציפים לא הכרחיים לא יהיו מורשים", כמו ההגיון אשר תומך בהנחה התרבותית "הנח את הגרוע מכל", ובאותו זמן סיפק לתלמידים נקודת מבט על ההשלכות של הרגלי העבודה שלהם ב"עולם האמיתי".

תפקיד המורה במפגש התרבותי

הסיטואציה הבאה מדגימה את התפקיד החשוב של המורה במפגש התרבותי. תלמיד מסר תדפיס של תכנית אשר היתה שגויה מכיוון שלא מילאה את אחת מהדרישות. המורה הבחינה מיד בטעות, אך בחרה לא לתקן אותה בצורה מפורשת. במקום זאת, היא פשוט שאלה את השאלות הבאות:

סיכום

התיאוריות הסוציו-תרבותיות אינן מספקות אלגוריתם לשיפור פדגוגיה. עם זאת, אנו רואים בהן מסגרות מאוד פוריות שבתוכן אנו יכולים לנתח קשיים פדגוגיים במדעי המחשב ולעצב התערבויות של הוראה. תיאוריות סוציו-תרבותיות מדגישות לא את התוכן של הידע הקיים של התלמידים, אלא את הדרך שהם ניגשים לפעילויות חינוכיות: אם הם נותנים לגיטימציה וכיצד הם ניגשים אליהן.

אנו טוענים שחיי המורים אף קשים יותר מאשר אולי הונח: לא רק שעלינו להיות מסוגלים לחשוף את הידע הקיים של התלמידים, אלא עלינו להיות מסוגלים לשים את עצמנו בנעליים שלהם במטרה להעריך את הגישה התרבותית שהם מביאים לסיטואציות בית ספריות.

אנו טוענים שהמונחים התנגשות תרבויות ואזורים פוריים של מפגש תרבויות הינם רלוונטיים לקהילה הרחבה של העוסקים בחינוך המדעי. במאמר זה הדגמנו את הכח של תיאוריות סוציו-תרבותיות בתוך הנושא הספציפי של מדעי המחשב, אך, השפעת החברות של התלמידים בתרבות של משתמשים טכנולוגיים על לימודיהם לא מוגבלת למדעי המחשב, אלא סביר להניח שהגישה התרבותית של התלמידים משפיעה על הביצועים בנושאים אחרים, בייחוד בהתחשב בשימוש הגובר של טכנולוגיית מחשבים בפעילויות בית ספר.

נספח: בעיית הפילוסופים הסועדים

בעיית הפילוסופים הסועדים הומצאה על ידי E. W. Dijkstra. זו דוגמה לבעיית קטע קריטי, שהיא בעיה מרכזית במקביליות, הנוגעת להקצאה בלעדית של משאבים, אשר דרושים לישויות במערכת רב-ישויות.

חמישה פילוסופים מעבירים את חייהם במחשבה ואכילה לחלופין. הם ישובים בחדר מסביב לשולחן עגול. בין כל זוג פילוסופים השכנים זה לזה מונח מזלג יחיד. במרכז השולחן ישנה קערת ספגטי (אינסופית). כאשר פילוסוף רוצה לאכול, הוא חייב להחזיק את שני המזלגות. נימוסים טובים דורשים כי פילוסוף ירים מזלג אחד בלבד בכל פעם. פתחו אלגוריתם עבור הפילוסופים אשר עונה על הדרישות הבאות: (א) מניעה הדדית (לכל היותר פילוסוף אחד מחזיק מזלג כלשהו בזמן כלשהו), (ב) אין קיפאון (אם מספר פילוסופים רוצים לאכול, בסופו של דבר, אחד מהם עושה זאת), (ג) אין הרעבה (כל פילוסוף אשר רוצה לאכול בסופו של דבר יעשה זאת).

אין פתרון מספק לחלוטין לבעיה זו. פתרונות לבעיה זו כוללים: (א) פתרונות לא-סימטריים (שבהם פילוסוף אחד מרים את המזלגות שלו בסדר שונה מהקולגות שלו), (ב) פתרונות עם תקורה נוספת (שוער המבטיח שלכל היותר ארבעה פילוסופים ינסו לאכול במקביל), (ג) אלגוריתמים אקראיים. המחקר והמשיכה הפדגוגית של בעיה זו נובעים מהעומק של הנושאים המושגיים והטכניים, אשר ניתן לדון בהם בתוך מסגרת משעשעת זו.

מחידושי האקדמיה ללשון העברית

פתח שיח – log on, log in	אוטומציה – automation
פתחה – port	הדממה – Shutdown
קוקית – Cookie	חברת הזנק – startup
רשת סריג – grid network	מחשב בֶּרֶכִּים – laptop computer
תחזוקה עתית – scheduled maintenance	מצת – plug
תחיבה – Insert	מְשִׁתָּה – אינטרנט
תסדיר – Format	מִשְׁקָף מקדים – previewer
תפריט צף – Pop Up Menu	נָעַל שיח – log out, log off

לקוח מכתבה שפורסמה ב-Ynet בכתובת <http://www.ynet.co.il/articles/0,7340,L-2296494,00.html>

רשימת מקורות

9. Leach, J. & Scott., P. (1995). The demands of learning science concepts: Issues in theory and practice. *School Science Review* 76(277):47-52.
10. Leach, J. & Scott., P. (2002). Designing and evaluating science teaching sequences: An approach drawing upon the concept of learning demand and a social constructivist perspective on learning. *Studies in Science Education*, 38:115-142.
11. Pollack, S. & Scherz, Z. (2003). Supporting project development in CS—the effect on intrinsic and extrinsic motivation. *Eleventh International PEG Conference*, St. Petersburg, Russia.
12. Slezak, P. (2001). Is cognitive science relevant to science teaching? *Sixth International History, Philosophy and Science Teaching Conference*, Denver, CO. Available on CDROM from <http://www1.umn.edu/ships/hpst/proceed.htm>
13. Turkle, S. (1995). *Life on the Screen: Identity in the Age of the Internet*. New York, NY: Simon and Schuster.
14. Wu, C-C, Dale, N. B. & Bethel. L. J. (1998). Cognitive models and cognitive learning styles in teaching recursion. *SIGCSE Bulletin*, 30(1):292-296.
15. Wenger, E. (1998). *Communities of Practice - Learning, Meaning and Identity*. Cambridge, UK: Cambridge University Press.
1. Ben-Ari, M. (2001). Constructivism in computer science education. *Journal of Computers in Mathematics and Science Teaching*, 20(1):45-73.
2. Ben-Ari, M. (in press). Situated learning in computer science education. *Computer Science Education*, 2004.
3. Booth, S. (2001). Learning to program as entering the datalogical culture: A phenomenographic exploration. *Ninth European Association for Research on Learning and Instruction Conference*, Fribourg, Switzerland.
<http://www.pedu.chalmers.se/shirley/EARLI2001/Booth.pdf>
4. Brousseau, G. (1997). *Theory of Didactical Situations in Mathematics*. Dordrecht: Kluwer Academic Publishers.
5. Göötschi, T., Sanders I. & Galpin, V. (2003). Mental models of recursion. *SIGCSE Bulletin* 35(1):346-350.
6. Gries, D. (2002). Problems in CSE. *Conference on Innovation and Technology in Computer Science Education*, Aarhus, Denmark. Retrieved on 11.3.04, from <http://www.iticse2002.dk/conference/Talk/iticse2002.html>
7. Kozulin, A. (1998). *Psychological Tools: A Sociocultural Approach to Education*. Cambridge, MA: Harvard University Press.
8. Lave, J. & Wenger, E. (1991). *Situated Learning: Legitimate Peripheral Participation*. Cambridge: Cambridge University Press.

תכנות מונחה עצמים עם תחילת המאה ה-21

סקירת ספרות, חלק ראשון

עפרה ברנדס

האוניברסיטה העברית, ירושלים

פתח דבר

לפני כמעט שנתיים, עם תום לימודי התואר השני במחלקה להוראת המדעים באוניברסיטה העברית, הגשתי סקירת ספרות בנושא "תכנות מונחה עצמים עם תחילת המאה ה-21", לפרופ' כתריאל בארי ולד"ר מירה עופרן.

בסקירה השתדלתי לעמת את הגישות המציעות להתחיל את הוראת מדעי המחשב בתכנות פרוצדורלי לעומת הגישות התומכות להתחיל מתכנות מונחה עצמים, והרשיתי לעצמי לגבש ולהציג מעין עמדה אישית בנושא.

עקב הרלוונטיות של הנושא לדיונים ולהצעות הפיתוח הנידונות בימים אלו במסגרות ועדת המקצוע, הפיקוח על מדעי המחשב ועוד, אני מבקשת לפרסם את עיקרי הסקירה בתקווה שישמשו רקע וחומר גלם למורים המעוניינים להשתלב בשיח בנושא. מכאן יוכל כל אחד להרחיב ולהעמיק כרצונו.

יש לציין ששנתיים במקצוע מדעי המחשב הן זמן רב מאוד. יש שינויים והתפתחויות. ודאי שיש מחקרים חדשים שנעשו בתחום והם מציעים עמדות נוספות.

אינני מוסיפה כרגע דבר לעבודה אלא מפרסמת אותה במתכונתה המקורית. לבקאים בפרדיגמות השונות בהוראת המדעים יקרוץ בודאי לנתח ולהדגים את הדעות השונות המובאות בסקירה לאור הגישות השונות הנהוגות בתחום זה (מאז הבהביוריזם, דרך פיאזיה ואוזובל, ועד הקונסטרוקטיביסטים השונים). זוהי תחושה נכונה אך גם פיתוח זה הקיים לצד העבודה לא יבוא לידי ביטוי בפרסום זה המיועד לשקף את העמדות עצמן.

עקב אורכה של הסקירה, יפורסם הפעם רק חלקה הראשון המציג את המושגים הנידונים ואת הבעיה העומדת על הפרק. כדי לאפשר למעוניינים לפנות כבר עתה למקורות, מצורפים גם עיקרי הביבליוגרפיה המצוינת לאורך העבודה.

יצויין שיחידת הלימוד אליה מתייחסת הסקירה, "תכנות מונחה עצמים בשפת ג'אווה", הפכה ליחידה מוכרת הנלמדת ברחבי הארץ. על פי הוראת הפיקוח ניתן יהיה ללמוד בכל בתי הספר החל משנת הלימודים תשס"ה תוך מימושה בג'אווה או בשפת C# (סי-שרפ).

אשמח אם פרסום הסקירה יעודד דיונים ענייניים בנושא, ואם יובאו לתשומת לב כולנו מחקרים ומאמרים חדשניים בתחום.

מבוא

מדעי המחשב, הם מדע צעיר יחסית, שאינו מונה עדיין מאה ללידתו. למרות גילו הצעיר, יצר המדע מהפכה אדירה באורחות החיים של העולם כולו. עם תחילת המילניום החדש, הוא תופס מקום משמעותי ומהדהד בכל תחומי חיינו. גילו הצעיר של המדע קובע את אופי ההתפתחויות ודינמיקת השינוי בתכנון. ככל מדע חדש הוא שרוי בשנים אלו בשלבי גיבוש והתהוות. מחד, יצירה וגיבוש של התיאוריות העומדות בבסיס המדע, ומאידך ובעיקר, פיתוח טכנולוגיות ישומיות המתרגמות את התיאוריות לכלים ואמצעים טכניים.

מהירות ההתחדשות בשנותיו הראשונות של מדע היא אדירה. בתוך פחות ממאה הפכו מדעי המחשב לטכנולוגיה השלטת וההתפתחות ממשיכה בקצב מדהים.

כשלושים שנה לאחר לידתו, קלטה והטמיעה מערכת החינוך, את המקצוע החדש לתוך מערכת ההוראה, ברבדיה השונים. הוחל בהוראת התחום בכיתות התיכון הגבוהות, בחינוך הגבוה והטכנולוגי, ולאחר מכן אף החלה הוראה של תחומי ידע קרובים (בעיקר שימושי מחשב) בכיתות בית הספר היסודי.

ההתפתחות המהירה של תחום דעת זה, היא בעלת השפעה משמעותית על החינוך למדעי המחשב, בתחום גופי הידע ובתחום הפדגוגיה.

תכניות הלימודים המנסות להקנות את עקרונות המדע ותכנון מתמודדות עם הצורך לייצר תכנית לימודים מקיפה ומשמעותית של התחום, תוך כדי מרדף בלתי פוסק אחר ההתקדמויות והגישות החדשות. קצב פיתוחן של אלו, עלייתן (ולעיתים גם דעיכתן), גבוה בהרבה מהקצב בו ניתן לגייס אנשים השולטים בתחום, לתרגם את הידע לתכנית לימודים, להטמיע את התכנית בקהילת המורים ובקרב אוכלוסית התלמידים.

בעייה זו של התחדשות, יחודית למדעי המחשב, צעיר המדעים הנלמדים היום במערכות החינוך.

בחזית היישומית במדעי המחשב עומדת בשנים האחרונות פרדיגמת התכנות מונחה העצמים (Object Oriented Programming).

פרדיגמה זו היא תוצאת התקדמות החשיבה וההבנה של פרדיגמות שקדמו לה והיא מהווה בסיס לפיתוחים טכנולוגיים רבי עוצמה בכל תחומי חיינו. בשלב זה היא מהווה כבר את הבסיס לפרדיגמות חדשניות יותר.

מתוקף תפקידי, כראש צוות מדעי המחשב, במרכז להוראת המדעים, באוניברסיטה העברית בירושלים, נדרשתי לסוגיית מיקומן של פרדיגמות "ישנות" ו/או פרדיגמות "חדשות" בתכנית הלימודים במדעי המחשב. לעיתים אף מצאתי עצמי בעין הסערה של הדיון הקריטי על קביעת הפרדיגמה הנלמדת, כאשר השיקולים המושמעים על ידי הצדדים השונים מערבים שיקולים מקצועיים ופדגוגיים עם טיעונים השייכים יותר לתחום השיווק וחוקי היצע וביקוש. בתוך עשור אחד מצאתי עצמי עוסקת בפיתוח שתי יחידות לימוד מתקדמות. האחת השתמשה בפרדיגמה "קלסית" של תכנות פרוצדורלי, ולימדה תכנים מסורתיים מתוך חדשנות והטמעה של גישות הפשטה מתקדמות שהחלו להופיע בעולם מדעי המחשב. כמעט בתום שלב ההטמעה הראשון של יחידה זו, ובהפרש שנים קטן מאד, מצאתי עצמי עוסקת בפיתוח יחידה המתמקדת בגישה שבחזית מדעי המחשב: "תכנות מונחה עצמים", שבה ניתן היה להגשים ולממש רעיונות מתקדמים שנרמזו ביחידה הקודמת. כעת כבר היו בידי המורים והתלמידים כלים חדשניים למימושם. מתוך כך פניתי לחפש ולהגדיר את שיקולינו והחלטותינו בתוך ההקשר הכללי של הבעיה.

מאמר זה יסקור בקצרה ובאופן חלקי את מקומן המסורתי והמתחדש של פרדיגמות התכנות כפי שהוא בא לידי ביטוי בספרות המקצועית בתחום תכניות הלימודים במדעי המחשב.

המאמר יעמת את גישת התכנות הפרוצדורלית המסורתית¹, עם פרדיגמת התכנות החדשה יחסית, המובילה בתחום מדעי המחשב ומציגה גישה שונה ומתקדמת יותר לתכנון ולפיתוח תוכניות, ה"תכנות מונחה העצמים"², ותנסה למקד את הסקירה בטיעונים לאופן הרכבת תכנית לימודים נכונה במדעי המחשב, בתחום צר זה של שפות וגישות הנתמכות על ידיהן.

1. בשדה החינוך היא מיוצגת בדרך כלל על ידי השפות פסקל (Pascal) ו-C.

2. מיוצגת בשדה החינוך בעיקר על ידי C++, Java.

תכנות פרוצדורלי³

שלב נוסף בהפשטה היה הקישור שבין הנתונים והפעולות האפשריות עליהם – לכדי חבילה סגורה אחת שניתן להשתמש בה בעזרת ממשק חיצוני, ללא ידיעת המימושים הפנימיים שלה. דבר זה נקרא בשם: טיפוס נתונים מופשט (Abstract Data Type). שלב זה היה שלב ביניים במעבר מהגישה הפרוצדורלית לגישת התכנות מונחה העצמים.

מקובל להתייחס לשפת פסקל כאל השפה הפרוצדורלית הטובה ביותר להוראה מכיון שהיא נבנתה ופותחה לצורך הוראה. ספרי הלימוד הקיימים היום במערכת החינוך הרשמית בארץ, והמתייחסים ליחידות הלימוד הראשונות בתיכון, כתובים בפסקל. יחידות אלו מיועדות להקנות את החומר הנכלל בקורס "מבוא למדעי המחשב" ובקורס "מבנה נתונים". לצד שפה זו, יש בתי ספר המלמדים את קורס המבוא והמשכו בשפת C. שפה זו תופסת מקום מכובד בתחום התעשייתי, והיא שפה שימושית בשוק המחשבים, אם כי כשפת הוראה היא נחשבת כמסורבלת ובלתי ידידותית לתלמיד.

תכנות מונחה עצמים

השלב הבא בהפשטה היה, שבהנתן בעייה תכנותית המגדירה מערכת כלשהי, תפורק המערכת לישויות המשתתפות בתהליך. כל ישות שכזו, נקראה עצם (object), והיא איגדה בתוכה את התכונות (attributes) המאפיינות אותה וערכיהן, עם התהליכים האפשריים על ערכים אלו (methods). בצורת חלוקה זו של מערכת, יחידת החלוקה הבסיסית אינה תהליך, אלא אוסף תהליכים שמכיל העצם, עם ערכיו. עצמים מסוג מסוים הם מופעים (instances) של תבנית בסיסית המגדירה את מהותו האבסטרקטית של העצם. תבנית זו נקראת מחלקה (class). בין מחלקות קיימת היררכיה הנקראת ירושה (inheritance) והיא מאפשרת בניית מחלקות ספציפיות על סמך מחלקות קיימות ושימוש מתוחכם בסוגי טיפוסים שונים, הנקרא רב-צורניות (polymorphism).

לאורך שנות ההתפתחות של שפות התכנות, ניכרת מגמה מתמשכת של התקדמות משפה פשוטה אותה יכול המחשב להבין ולפעול על פיה אך היא רחוקה מאד מדרך החשיבה, הביטוי וההתנסחות האנושית, לכיוון שפות המביאות לידי ביטוי את השיח האנושי. שפות אלו מתרחקות יותר ויותר ממבנה שפת המחשב הבסיסית ומצריכות פלטפורמות תרגום ועיבוד (מהדרים ומפענחים) שיתרגמו את התכנית הגמורה למשימה ב"שפתו" של המחשב.

ההתקרבות לחשיבה ולדרך הביטוי האנושית משקפת את הצורך שעמד בבסיס המצאת המחשב: חישוביות וביצועיות המהירות ויעילות מכוח העשייה העצמאית של האדם. לצרכים אלו חייב האדם לעבוד בדרך הנוחה למבנה החשיבה והביטוי האנושי, ולא לבזבז אנרגיות בשירות המכונה.

שפת המכונה הראשונית שהורכבה מצרופים של אפסים ואחדים בלבד, "כוסתה" ברבדים של סמלים ותוויות שהתקרבו לדרך הכתיבה האנושית. לאחר מכן הוחלפה שפת המחשב בפקודות "אנושיות". על הרצף הסדרתי של צעדים בסיסיים הוגדרו מבני שליטה ובקרה (תנאים ולולאות). עם גידול התכניות והמעבר לטיפול במערכות גדולות היה צורך לגבש את תהליך הביצוע ולהופכו מאוסף עצום של הוראות לכלל תהליכים חוזרים המשתלבים אלו באלו.

בשלב זה פותחו השפות הפרוצדורליות. בהנתן משימה, היא מתפרטת ל**תהליכים** מרכזיים. הפעלתם של התהליכים (פונקציות ופרוצדורות), על הנתונים המתאימים ושילובם הנכון, באופן מבני ומאורגן – תביא לסיומו המוצלח של תהליך אלגוריתמי מוגדר והמשימה תגיע לסיומה. התוספת של פרוצדורות, שימשה כשלב בהפשטה של תהליכים ומשימות (אבסטרקציה). במקום לטפל בנתונים בצעדים בסיסיים בודדים, התאפשר להגדיר אוסף של צעדים שכאלה כצעד בסיסי חדש.

3. ההגדרות בשני הסעיפים הבאים פשטניות ומיועדות רק לצורך קביעת שפת דיבור מובנת לכול בקריאת עבודה זו.

איזו פרדיגמה תחילה?

הצגנו שתי פרדיגמות תכנותיות שונות.

האם ההבדל ביניהן הוא עקרוני? או שמא העניין הוא כרונולוגי, פרדיגמה אחת היא פיתוח של הקודמת לה וזה הסדר ביניהן?

האם יש עדיפות באיזו פרדיגמה להתחיל את הוראת מדעי המחשב?

האם יש הכרעה מקובלת ומוסכמת בדבר עדיפות זו?

האם התחלה בפרדיגמה כלשהי היא בעלת השפעה על לימוד הפרדיגמה האחרת? ואיזו השפעה?

והאם קביעת הפרדיגמה מושפעת ממטרת ההוראה, ולמטרות שונות יתאימו פרדיגמות שונות?

כדי לענות על כל השאלות הללו ומעבר לכך, יש לקיים השוואה עניינית בין שתי גישות התכנות, הנוגעת לעקרונות שלהן, לסדר הלימוד וההבנה שלהן, ולראות אם יש הכרעה מוכחת ומקובלת בעולם החינוך למדעי המחשב. נסקור רק מעט מזעיר מהמאמרים והדיונים סביב נושא זה. למרות שעיקר עוצמתו של הדיון באמצע שנות התשעים, נתמקד במאמרים המאוחרים יחסית שפורסמו מאמצע שנות התשעים ועד ראשית המאה העשרים ואחת, מתוך שהם מסכמים את מה שנאמר לפנייהם. עיון מהיר מגלה שקיימת הטייה גדולה לטובת פרסומים וטיעונים מן הסוג של "תמ"ע עדיף", הרבה פחות טיעונים לאי-שוני בין השיטות⁶, ואמירות אגביות בלבד בדבר האופציה של "פרוצדורלי עדיף עדיף".

ניתן היה לכאורה לסכם את הקשר בין הפרדיגמות בכרונולוגיה פשוטה: הגישה הפרוצדורלית היא הקלסית. היא קדמה לתמ"ע, השתרשה בספרי הלימוד ובתוכניות הלימודים והיא שגורה בפי המורים. התכנות מונחה העצמים הוא נדבך נוסף וחדש שעדיין לא בא לידי ביטוי בספרי הלימוד הקיימים. ואמנם יש אנשי חינוך⁷, הטוענים שעדיין יש להתחיל את הלימוד של מדעי המחשב מהגישה הפרוצדורלית ורק לאחר מכן לחשוף את התלמידים לתכנות מונחה העצמים.

התקשורת האפשרית המתקיימת בין עצמים מאפשרת להקים ולשלוט במערכות גדולות מאד, על ידי צוותים רבי משתתפים שאינם בהכרח פועלים ביחד על כל יחידה, ואינם מכירים היכרות פנימית זה את עבודתו של זה. רמת האבסטרקציה והסתרת המידע בתוך העצמים והמחלקות, אפשרה עבודה עם מחלקות קיימות דרך הממשקים שלהן. ממשקים אלו חשפו למשתמש רק את המידע ההכרחי על דרך השימוש בעצם, ללא העמסת פרטי מידע מיותרים על דרכי המימוש וההפעלה הפנימיים. נוצר אוסף עצום של מחלקות העומדות לרשותו של כל משתמש, בדוקות ובטוחות וניתנות לשימוש ושילוב במערכות מתקדמות, תוך חיסכון בכוח אדם, ומתוך בטיחות מקסימלית של ביצועים קיימים.

בין השפות המקובלות במערכות החינוך השונות להוראת תכנות מונחה עצמים, יש להזכיר את ג'אווה (Java) שפותחה מראשיתה כשפה מונחית עצמים, היא נקיה מאד וידידותית ללומד.

שפת OOP⁴ אחרת, המקובלת מאד בתעשייה כיום, היא ++C ולכן רבים מבתי הספר הטכנולוגיים בחרו בשפה זו. היא מורכבת ומסובכת יותר ללימוד ונחשבת קשה יותר מג'אווה ללומד המתחיל. בכל זאת יש בתי ספר המתחילים את קורס המבוא בשפת C ומשם הדרך הטבעית היא להמשיך בלימוד ++C כשפה ממשיכה⁵.

בבתי הספר העל-יסודיים בארץ, עדיין לא מלמדים באופן נרחב את הפרדיגמה הנידונה, אך במספר מצומצם של בתי ספר נלמדת חלופה ליחידה החמישית במדעי המחשב, בשם תכנות מונחה עצמים. חלק מבתי הספר מלמדים על פי ספרים שפותחו בשוק הפרטי, בשפת ++C, ומספר מצומצם של בתי ספר משתתף זו השנה השלישית בניסוי של החלופה שפותחה באוניברסיטה העברית בירושלים, והמיושמת בג'אווה. כמו כן, ראוי להזכיר שהחל משנת תשס"ז מתוכנן מעבר לג'אווה וסי שרפ גם ביסודות ועיצוב תכנה.

6. בסקירה זו ייוצג על ידי המחקר ההשוואתי של ווידנבק, ראו מקור 8.

7. גיבונס [5], כמייצג דעה של רבים אחרים.

4. תכנות מונחה עצמים יקרא להלן כך, או תמ"ע, או OOP, לחילופין ובאופן חד ערכי.

5. "דעת יחיד" שמצאתי עד כה, היוצאת חוצץ נגד הוראת תכנות מונחה עצמים ב-++C לתלמידים יודעי C, היא של ברגין, וארחיב על כך בהמשך.

הקינן של פונקציות ופרוצדורות כפי שמציעות אותן השפות הפרוצדורליות (בעיקר פסקל), תומך בדיוק בצורת פתרון בעיות שכזו, ומאפשר הסתרת מידע ואנקפסולציה¹⁰. פתרונות שכאלו נראים כמו עצים בהם כל צומת הוא תחילתה של פונקציה או פרוצדורה. בצורת תכנות שכזו יש למזער ככל האפשר את כמות האינפורמציה העוברת בין התהליכים השונים ולוודא שכל האינפורמציה המועברת מאורגנת על גבי הקשרים בתוך העץ (בצורת פרמטרים הנשלחים לתהליכים). התהליכים הנמצאים בעלי העץ אינם מקושרים לשום הליך נוסף. אם בתוך כדי תהליך התכנון של פתרון, מסתבר שאי אפשר לקשר תהליכים או שאין דרך להעביר כראוי את האינפורמציה, יש "לפרוס" חלק מהעץ ולהגדיר את התהליכים שבו מחדש. לאוסף אלגוריתמים זה יש להוסיף את הנתונים הדרושים ואז על פי הגדרתו של ניקולאוס ווירט¹¹, נקבל תכנית.

החשיבה בעת עיצוב הפתרון היא קודם כל על האלגוריתמים ולא על הנתונים. אלו מצטרפים ברגע האחרון ומונעים למעשה הסתרת אינפורמציה מושלמת.

בתכנות מונחה עצמים החשיבה הפוכה ממש. איננו חושבים על תכנית כתהליך ואיננו חושבים על פירוק משימה למשימות קטנות. ההתבוננות הראשונית בתמ"ע היא על "הנתונים" ולא על האלגוריתמים המעורבים במשימה. אנו מתייחסים אל האובייקטים כנתוני המערכת. האובייקטים כומסים בתוך עצמם את האלגוריתמים ואת הנתונים הפשוטים (data). תכנון העצמים עצמם הוא מהלך משני. אוסף העצמים המתקשרים ביניהם נותן מבנה של גרף או רשת בה יש קשר רב-מימדי בין הרכיבים השונים. שלב התכנון של פתרון בפרדיגמה זו מתחיל בזיהוי האובייקטים המשתתפים בתהליך הרצוי (לרוב דרך סימולציה לעולם הבעיה האמיתית).

גישה זו רואה בתכנות מונחה העצמים נדבך נוסף והרחבה של הגישה הפרוצדורלית וממילא רואה את דרך ההוראה כמקבילה לסדר התפתחות השפות. להצגת הדברים זהו יש מתנגדים נמרצים⁸ הטוענים שגם אם כרונולוגית השפות הן התפתחות זו של זו, הרי שהתכנות מונחה העצמים אינו פיתוח והרחבה של התכנות הפרוצדורלי. התכנות מונחה העצמים מייצג גישת חשיבה השונה לחלוטין במהותה ואין ליצר רצף התפתחותי בהוראת השפות ובדאי לא רצף התואם את הרצף הכרונולוגי (בדיעבד הם יטענו שהרצף היחיד האפשרי, הוא הרצף ההפוך).

נקודת מוצא המוסכמת על כלל המתדיינים היא שההוראה הקלסית נוחה למורים, וכי כל מעבר לשיטה חדשה מצריך כוחות הוראה והשקעת משאבים ומאמצים רבים. הצורך לכתוב תוכניות לימודים חדשות, להתאים מחדש את רצף הקורסים על פי התכנים החדשים, להגיע למצב בו נוח למורים בתוך הסביבה החדשה כמו שהרגישו בסביבה המוכרת להם, הוא תהליך קשה ולא תמיד מאתגר. התומכים בביצוע המעבר משתדלים להציע ספרות תומכת, לחלוק רעיונות מקוריים ומוצלחים להצגת תכני הקורס, כדי להקל על המעבר, אך הבעיה מובנת.

במאמרו "מדוע הגישה הפרוצדורלית היא הפרדיגמה השגויה להתחלה אם תכנות מונחה עצמים הוא המטרה" טוען ברגין שאם תכנות מונחה עצמים הוא המטרה – אין להתחיל ממשוה אחר, והוא אף מציע גישה לקורס התחלה. גם אם נקבל את עמדותיו תשאר השאלה בעינה: ואם לא התמ"ע הוא המטרה אלא לימוד מדעי המחשב לעצמו הוא המטרה, מה אז? על כך משתדלים לענות דקר והירשפילד במאמרם "עשר סיבות למה תכנות מונחה עצמים לא יכול להלמד בקורס מבוא למדחמ"ח"⁹.

ברגין מגדיר את עיקר המהות בגישת התכנות הפרוצדורלי, באופן בו היא ניגשת לפתרון בעיות. עיצוב הפתרון נעשה בגישת ה-Top-Down Design. בהנתן בעייה יש לחלק אותה לתת-משימות (וחזור חלילה), עד למצב בו לכל תת-משימה ניתן להציע פתרון. בהנתן הפתרונות החלקיים לכל תת-המשימות, נפתרת ממילא בעיית המוצא.

8. בעבודה זו ייצג אותם ברגין [2].

9. מקור [3].

10. לטענתו, C – שאינה מאפשרת קינון פונקציות בדרך זו, אינה תומכת כראוי בנושא הסתרת מידע ואנקפסולציה.

11. Niklaus Wirth ממציאה של שפת פסקל ושפות פרוצדורליות נוספות, טען ש-"אלגוריתם+נתונים = תוכנית".

טיעונים אלו יכולים להוביל להמלצות אופרטיוויות שלא להתפס להקלות קיצוניות בהוראת התכנות או לא להתחיל את לימודי המחשב בתכנות כלל, אך הם אינם מקבעים פרדיגמת התחלה. טיעוניו של ברגין לעומת זאת מחייבים את ההתחלה מתכנות מונחה עצמים. מסקנתו של ברגין נראית סותרת את יכולת הטבע האנושי ללמוד, להתפתח ולהחליף ידע קיים בידע חדש, מתקדם יותר ואף להפוך אותו לטבע החשיבה באופן מודע.

לחיזוק גישתו שיש להתחיל בגישת התמ"ע, מצטט ברגין ממצאים של מחקרים מעשיים שונים. ראשית, לא קשה יותר ללמד לכתחילה את הפרדיגמה של תמ"ע מאשר כל פרדיגמה תכנותית אחרת¹³, אך כדי להעביר מתכנת טוב מחשיבה פרוצדורלית לחשיבה מונחית עצמים נדרשים לפחות בין שנה לשמונה-עשר חודשים¹⁴. ברגין אינו מוצא במהלך ארבע שנות לימוד, זמן "לבזבז" על המעבר בין הפרדיגמות.

מתוך מחקר נוסף¹⁵ הוא מדווח על קשיים שונים שיש בהוראת תמ"ע לתלמידים השולטים בגישה הפרוצדורלית:

על פי המחקר, ברגע שהמשימה מסתבכת, המתכנת יסוג חזרה למה שידוע לו טוב יותר כלומר לגישה הפרוצדורלית. נדרש זמן רב כדי לשנות את הרגלי החשיבה.

במהלך זמן לימוד זה יכתבו התלמידים תוכניות "מכוערות" בהן יבוא לידי ביטוי ערבוב של הגישות השונות.

נסכם את ההבדל בין הפרדיגמות בגישה לפתרון בעיות:

המתכנת בתמ"ע מחפש ומגדיר אובייקטים המייצגים דרך התנהגות. לאחר מכן הוא מגדיר את השירותים שהאובייקטים יכולים לתת (אובייקט מאובחן שלא יהיו לו שיטות שכאלה, אינו נדרש למעשה, בעוד שמספר אובייקטים בעלי אותן שיטות יגדירו מחלקה אחת). ורק לבסוף הוא מחליט עבור כל אובייקט מהם הנתונים המגדירים אותו (ערכי התכונות). במידה וימצאו נתונים המשותפים לכלל האובייקטים הם יוגדרו בתוך המחלקה (static).

לעומתו, המתכנת הפרוצדורלי מתחיל את הגדרת הפתרון מהשלב השני של הגדרת התהליכים. והנתונים אינם מהווים לגביו חלק אינטגרלי מהבעיה.

הבחנה יסודית ומפורטת זו הובאה כדי לתת לקורא מובן מסוים על עומק השוני המחשבתי הבא לידי ביטוי בפרדיגמות השונות. הכרעה על פרדיגמה פירושה למעשה העדפת צורת חשיבה אחת על גבי האחרת.

פרט להבחנה מעמיקה ויסודית זו בין הפרדיגמות, מגיע ברגין לדיון מצויד בהנחות יסוד על טבע החשיבה של האדם. ברגין מניח שאדם יעבוד טוב ביותר בפרדיגמה התכנותית הטבעית לו. תהליך החשיבה הטבעי יגרום לאדם להעדיף את מציאת הפתרון בדרך המוכרת לו מאשר בדרך המתאימה ביותר לרוח הבעיה. מכיון שהמטרה אצל ברגין היא ידיעת התמ"ע, ברור שהוא יעדיף את לימוד ה-OOP כפרדיגמה ראשונה, וכך בכל מצב "יסוג" המתכנת לפרדיגמה זו בזמן חיפוש הפתרון.

במאמרים רבים הדנים בסוגיית הקניית מיומנויות תכנות בראשית לימודי המחשב, ניתן למצוא תמיכה מסוימת במסקנותיו של ברגין. במסגרת הטיעונים נגד הקניית מיומנויות תכנות (כלשהן!), מוצגת טענה דומה בדבר הנסיגה להרגלי למידה ראשונים, אך המקור לטיעון שונה. המאמרים מדברים על המצב המקובל בו מפשטים את תהליך התכנות עבור התלמיד המתחיל, ומסתפקים במינון נמוך של תהליכי ניתוח ותכנון. התלמיד גומר את לימודיו עם רושם מוטעה על מהות התיכנות ומתקשה בהמשך לנתח מצבים מורכבים או לאמץ גישות מתקדמות ושונות לפתרון בעיות¹².

12. דיעה זו באה לידי ביטוי גם בתכנית הקרויה CC2001.

13. דיעה החלטית זו אינה מציינת את מסקנותיהם ההפוכות של מחקרים מעשיים אחרים, ש-CC2001 מתבסס עליהם וכן הדו"ח של מכון ויצמן, הטוען לקושי ברמת ההפשטה הנדרשת להבנת תמ"ע.

14. מקור [7].

15. מקור [6].

למרות רמת ההוראה הגבוהה ששכללו המורים בקולג', התוצאות בשנים האחרונות הראו שהבוגרים חסרים את המיומנויות הנדרשות מכל מהנדס תוכנה. לצד חוסר נחת זה מתווספת פליאה מרמת ההפשטה והביצועים שהפגינו תלמידי חוץ שלמדו קורס יחידאי (ללא כוונת המשך במדהמ"ח) ובו עבדו בגישה דמוית תכנות מונחה עצמים¹⁷.

התחושה היתה שהתשובה נמצאת בפרדיגמה של תכנות מונחה עצמים. והתשובה אכן נמצאה באימוץ השיטה החדשה. הפרדיגמה החדשה, להבדיל מהאופן בו רואה אותה ברגין, **הרחיבה** את הגישה המוכרת של התכנות הפרוצדורלי ונתנה מענה לכל הבעיות שהתעוררו בקולג'. במקום ליצור גודש ועומס על מערכת ההוראה העמוסה בלאו הכי, נוצר קישור טבעי בין נושאי לימוד ותכנים שונים, והגישה התאימה החל מקורס המבוא ועד תכנים מתקדמים בקורסי ההמשך. הוראתה בקורס המבוא חסכה את פער המעבר המתואר גם אצל ברגין ובמחקרים נוספים, ואפשרה לתלמידים להחשף ישירות לגישה המתאימה במיוחד לטיפול במערכות אמיתיות-מורכבות בעלות ממשק משתמש מורכב.

בסיכום מאמרו, מציע ברגין מבנה של קורס מבוא בתמ"ע¹⁸ בו נחשף התלמיד למושגי היסוד (עצם, שיטות, מחלקה, ואפילו ירושה) ולרעיונות היסוד (תקשורת בין עצמים) הרבה לפני שהוא כותב שורת קוד ראשונה. החשיפה נעשית תוך הפעלת עולם של צפרדעים שבו התלמיד מפעיל את האינטרקציה בין עצמים=צפרדעים¹⁹.

16. יש אקדמיות בארץ המבצעות הלכה למעשה את מסקנתו של ברגין: קורס המבוא נלמד ומיושם בג'אוה, ובשנה ב' לומדים התלמידים את הקורס "מעבדת תכנות" ובו הם נחשפים לשפת C כמייצגת את הגישה הפרוצדורלית. נציין, שלא מעט סטודנטים המדברים בעגה מונחית עצמים ומשתמשים במושגי אובייקט ומחלקות מיום כניסתם לאוניברסיטה, העידו על עצמם שלא הבינו את מהות המושגים ועומקם עד שלא למדו C! חלק אחר ראה בכך גזירת האוניברסיטה ולא ראה שום צורך לחזור לשפה שכזו.

17. HyperCard; HyperTalk.

18. לקוח מתוך קורס המועבר באוניברסיטה הפתוחה באנגליה.

19. בחלק מהשנים נלמד קורס המבוא באוניברסיטה העברית באופן דומה ממש כאשר במקום דוגמת עולם הצפרדעים שמציע ברגין, משתמש פרופ' רוזנשיין בדוגמת עולם הרובוטים.

מכיון שכל השפות המונחות עצמים מכילות בסיס פרוצדורלי בבסיסן, וחלקן (כמו C++) אף מאפשרות להמנע משימוש בעצמים, יכול התלמיד לעבור פורמלית לתכנת בשפה מונחית עצמים אך למעשה להמשיך בה את התכנות הפרוצדורלי עליו הוא אמון. כתוצאה מממצא זה מציג ברגין דיעה יחידאית הטוענת שלמרות הסינטקס הדומה בין C לבין C++ עדיף שלא להתקדם מזו לזו שכן המעבר לא יתמוך בשינוי דרכי החשיבה והרגלי התכנות.

כמסקנה מסכמת, מעדיף ברגין את הלימוד ההפוך: ללמד את הגישה הפרוצדורלית לתלמידים השולטים ב-OOP. באופן שכזה, ימשיכו לאתר ולהגדיר יחידות דמויות עצמים מבחינת הפונקציונליות שלהן, למרות שלא יוכלו להמשיך ולהשתמש בהסתרת המידע והכמסתו כפי שיכלו לעשות בתמ"ע¹⁶.

דקר והירשפילד מגיעים למסקנות מאד דומות אך מנקודת מוצא שונה. אין הם מגדירים שתמ"ע הוא המטרה בהוראתם. במאמרם הם מתארים את הלבטים עימם התמודדו עד שהגיעו להחלטה (עליה הם מאד שמחים) לשנות את קורס המבוא בקולג' שלהם. במשך שנים הועבר קורס מבוא למדהמ"ח בגישה הפרוצדורלית, ויושם בפסקל. מאמרם מציג את עשרת הטיעונים העיקריים אותם העלו בכל התקופה בה ניסו לשכנע את עצמם שלא לבצע את המעבר, ומה השיבו למעשה לכל אחד מהטיעונים, טרם המעבר ולאחריו.

עשרת הטיעונים הם הטיעונים המקובלים המוצאים ביטוי בכל המאמרים העוסקים בנושא. בסכמם את הטיעונים הם אומרים, שכל הטיעונים שיקפו למעשה את חוסר הבקיאיות וההבנה המעמיקה שלהם כמורים את הפרדיגמה החדשה, את הפחד מהשפה והחידוש, את הידע, השליטה והנוחות העצומים שהרגישו כמורים בהוראה המקובלת והשגרתית.

המוטיבציה לשינוי אצל דקר והירשפילד היא מתוך חוסר נוחות ואכזבה מתוצאות ההוראה ורמת התלמידים בגישה המקובלת. תלמידי המחלקה הפגינו חוסר בכישורים בסיסיים וביכולות הפשטה ותכנון שהיו דרושים לצורך תכנות מתקדם בהמשך.

מסתמנת נטייה ברורה למעבר לפרדיגמה החדשה גם באותם מקומות בהם מדווחים קשיים בהנחלת מושגי היסוד לתלמידים. הדיון במקרים אלו מתמקד לא בערעור על רעיון ההחלפה אלא בהצעת קורסים חילופיים ומקוריים המתקיפים את הנושא מכיוונים שונים.

לסיכום הדיעות המצדדות באימוץ הגישה מונחת העצמים בקורסי המבוא, נאמר שהחידוש שנעשה בכל תחומי התכנות בעולם הסובב את האקדמיה וגם את התיכון, והנשען על מושגי העצמים, הקרין, חייב ואף הצדיק מעבר לתכני לימוד חדשים. קשה היה להמשיך ולתמוך בשימור הגישה הישנה בעוד שבעולם התעשייה והשימושים הביתיים (אינטרנט) אליו נחשפו התלמידים ומוריהם, התבצע שינוי רב משמעות שכווה. האפשרות של הוספת תכנים לאחר הוראת הגישה המסורתית, נבחנה ונפסלה במקרים רבים עקב העומס על תכנית הלימודים ועקב בעיות מושגים והבנה של התכנים החדשים, שנצפו כלא מתקיימים אם משנים את סדר ההוראה של הפרדיגמות.

האם הגישה הזו מוחלטת?

בפרקים הבאים נסקור מסמכים רשמיים הסוקרים את הנימוקים והשיקולים בעיצוב תוכניות לימודים בארץ ובעולם. נראה שההחלטות המשתקפת במאמרים שלעיל, אינה מקובלת שם, וכי מעצבי תוכניות הלימודים רואים עדיין מקום בקיום הגישה המסורתית הקיימת כולל תוכניות הלימודים שפותחו על פיה.

מקורות

1. Computing Curricula 2001 Computer Science/ Final Report December 15, 2001 *ACM Journal of Educational and Resources in Computing*, Vol. 1, No. 3, Fall 2001 (240 p.)
2. Bergin, J.(2000).Why Procedural is the Wrong First Paradigm if OOP is the Goal. Presented at *OOPSLA99 educator's symposium*.
<http://csis.pace.edu/~bergin/papers/Whynotproceduralfirst.html>

תוכן הקורס – אינו מעניינה של סקירה זו. החשוב לגביו הוא שאכן ברגין דוחה את רוב העיסוק בבסיס הפרוצדורלי הקיים וההכרחי בשפה (לולאות, תנאים מקוננים, switch וטיפוסי נתונים בסיסיים) לשלבים מאוחרים בהרבה ובמקום זאת שם את הדגש על שימוש בעצמים, הפעלת ירושה על מחלקות קיימות, לימוד רקורסיה באופן טבעי של מסרים בין עצמים לעצמם. ברגין אכן עקבי בגישתו, בעיסוק היסודי והמקדמי באושיות הפרדיגמה טרם העיסוק בבסיס הפרוצדורלי הקיים בה. גישתו זו אינה הגישה המקובלת אם כי היא ראויה אולי להיות מומלצת ומאומצת.

מחקרם ההשוואתי של ווידנבק ושות' [8] נערך ב-1997 ועודכן בשנה שלאחר מכן. במסגרת המחקר התבצעו שני ניסויים שמטרתם לבדוק את ההבנה של תוכניות קצרות ותוכניות ארוכות, על ידי תלמידים שלמדו קורס תכנות מתקדם בגישה הפרוצדורלית לעומת אלו שלמדו בגישת התכנות מונחה העצמים. החוקרים קבעו מדדי השוואה ובדיקה לרמת ההבנה של התוכניות על ידי התלמידים. בסיכום הממצאים הם מודים שלא נמצאה הכרעה משמעותית לצד כלשהו. אם בתוכניות הקצרות היתה עדיפות בהבנת הפונקציונליות של התוכנית אצל יודעי התכנות מונחה העצמים, הרי שהיא התבטלה בהבנה יסודית וטובה יותר של יודעי התכנות הפרוצדורלי בתוכניות הגדולות, ועוד. סך כל הממצאים לא הראו שונות בין ההבנה הכוללת של הקבוצות. החוקרים מציעים להמשיך את המחקרים לכמה כיוונים אך התוצאה המעשית בשטח ברורה: אין הכרעה חד משמעית בין השיטות השונות.

ככל שעוקבים אחר הפרסומים והמאמרים בעולם מדעי המחשב בשנות התשעים הולך ומסתמן קו אחיד של מחלוקת שהוכרעה. מכיוונים שונים מגיעות תמיכות לרעיון ההחלפה של פרדיגמת ההוראה בקורסי המבוא. הכותבים רק מתחרים ביניהם בטיעונים שהם מעלים כדי לתמוך בשינוי, ומידי פעם אף נחלקים בדבר השפה המועדפת להוראה בשיטה החדשה.

8. Wiedenbeck, S., Ramalingam, V., Sarasamma, S., Corritore, C.L. (1998).
9. A Comparison of the comprehension of object-oriented and procedural programs by novice programmers, *Interacting with Computers 11* (1999) 255-282.
3. Decker, R., Hirshfield, S. (1994). The Top 10 Reasons Why Object-Oriented Programming Can't Be Taught in CS1. *Selected papers of the 25th annual SIGCSE symposium on CSE, March 1994 SIGCSE Bulletin, Vol.26, No.1*, 51-55.
4. Gal-Ezer, J., Beeri, C., Harel, D., Yehudai, A.(1995). A high School Program in Computer Science. *COMPUTER – Innovative technology for computer professionals, October 1995*, 73-80.
5. Gibbons, J.(1998) Structured Programming in Java, *SIGPLAN Notices, Vol. 33, No.4*.
6. Lattanzi, M., Henry, S.(1996) Teaching the Object-Oriented Paradigm and Software Reuse *Computer Science Education, Vol.7, No.1*, 99.
7. Stroustrup, B. (1994) The Design and Evolution of C++, *Addison Wesley*.
10. רגוניס, נ., בן-ארי, מ., לוי-רשתי, ד., רייך, נ. (2001). מודרניזציה של קורסים במדעי המחשב. *דו"ח מסכם, המחלקה להוראת המדעים, מכון ויצמן למדע*.
11. תכנות מונחה עצמים בשפת ג'אווה (2002) *צוות מדעי המחשב, המרכז להוראת המדעים, האוניברסיטה העברית בירושלים*.
12. אתר הפיקוח על מדעי המחשב וטכנולוגיות מידע: <http://www.csit.org.il/main/menu.asp>

מקורות בעברית

קול קורא

השתלמות בתכנות מונחה עצמים בשפת ג'אווה

מועברת על ידי צוות הפיתוח, האוניברסיטה העברית בירושלים

על פי חומרי הלימוד שפותחו לחלופה ליחידה החמישית (באישור הפיקוח)

החל משנת הלימודים תשס"ה יתאפשר לכל מורה המעוניין בכך, ללמד את החלופה החדשה ליחידה החמישית: "תכנות מונחה עצמים בשפת ג'אווה". היחידה שפותחה באוניברסיטה העברית בירושלים, נוסתה בהצלחה במשך שנתיים ונלמדת זו השנה השלישית במספר בתי ספר.

החל משנת תשס"ז מתוכננים מספר שינויים בתכנית הלימודים, במסגרתם יוחלפו שפות הלימוד הקיימות לשפות מונחות עצמים. השתלמות בתחום תאפשר למורים להתוודע לפרדיגמה החדשה ולהתחיל להתרגל לשינויים הצפויים בכל מערך ההוראה.

צוות הפיתוח של היחידה בודק את הצורך והיכולת להעביר השתלמויות נוספות על אלו שהועברו עד כה. עלותה של השתלמות שתועבר על ידי הצוות המקורי אינה מבוטלת, ולכן תפתח השתלמות רק במידה ומספר הנרשמים באזור מסוים יצדיק זאת.

אורך השתלמות מלאה הוא 84 שעות, במסגרתן מועברים גם פרקי העשרה למורים.

ניתן לקיים השתלמות מקוצרת באורך 56 שעות, ללא ההעשרה.

במהלך ההשתלמות יחשפו המורים לכל חומרי הלימוד שפותחו באוניברסיטה העברית: הספר לתלמיד, המדריך למורה, סביבת העבודה וקבצי העזר הנלוים אליה, מערך המצגות המלא המלווה את היחידה ומאפשר למורה להורות בקלות את כל היחידה.

כמו כן יבצעו המשתתפים את דפי העבודה בהנחיה צמודה של הצוות ויפתרו מבחנים ותרגילים נוספים.

המעוניינים ישלחו בהקדם את שמם, שם בית הספר בו הם מגישים לבגרות ביחידת "עיצוב תוכנה" (מנסיונו – רצוי לעבור את ההשתלמות רק לאחר היכרות טובה עם היחידה הרביעית), את המקום בו הם מבקשים לקיים את ההשתלמות וכן את משך ההשתלמות הרצוי להם, לכתובת: brandes@huji.ac.il

שימוש בחידות כאמצעי לשיפור מיומנויות של פתרון בעיות אלגוריתמיות

שרה פולק וד"ר צביקה פירסט

המרכז לטכנולוגיה חינוכית (מט"ח)

מבוא

אחת המטרות העיקריות של מערכת החינוך בכלל ושל הוראת מדעי המחשב בפרט היא להקנות לתלמידים מיומנויות לפתרון בעיות, ואכן הפעילות העיקרית בה עוסקים תלמידים במדעי המחשב היא פתרון בעיות אלגוריתמיות. פתרון בעיות אלו מסייע לתלמידים להכיר את המושגים הבסיסיים במדעי המחשב כמו אלגוריתם, יעילות, נכונות, שפת תכנות ועוד.

אם נבחן את סוג הבעיות המוצגות בספרי הלימוד ובבחינות הבגרות במדעי המחשב, נראה כי ניתן לאפיין כבעיות הלכות מעולם הפתרון. במילים אחרות, הבעיות בדרך כלל כוללות סיפור פשוט וברור של עולם הבעיה והדגש הוא על כתיבת תכנית הכוללת שימוש בתבניות תכנות שונות. לדוגמה, שתי הבעיות הבאות (שהופיעו בבחינת בגרות, שאלון 899222, תשס"ג) לקוחות מעולם הפתרון:

- נתון מערך חד-ממדי a בגודל 27, המכיל מחרוזות שונות זו מזו. כתוב קטע תכנית שיקלוט מחרוזת וידפיס את מיקומה במערך.

- בגן משחקים יש כניסה המובילה לשני מבוכים big ו-small. המבוך big מיועד לאנשים שגובהם מ-1.70 מ' ומעלה, והמבוך small מיועד לאנשים שגובהם פחות מ-1.70 מ'. כתוב תכנית שתקלוט את גבהי האנשים ותציג כמה אנשים נכנסו בכל מבוך...."

לשימוש בבעיות שרובן לקוחות מעולם הפתרון, יש השלכות על הפרקטיקה והמיומנויות של פתרון בעיות שתלמידים רוכשים במדעי המחשב.

מאחר והבעיות הן "פשוטות" והעיקר בהן הוא השימוש והיישום של תבניות תכנות מתאימות, תלמידים רבים מרגישים כי הם יכולים לעבוד "ישר על המחשב", על-ידי כתיבת רעיון ראשוני ובחינתו בניסוי וטעייה, ללא ניתוח מעמיק ולעיתים קרובות גם ללא תכנון ראוי של הפתרון. בהמשך, תלמידים אלו "מסרבים" לסגל שיטות מתאימות לפתרון בעיות, ומתקשים להתמודד עם בעיות "שחורגות" מהסטנדרט המקובל. כלומר, בעיות שבהן הפתרון של הבעיה אינו מיידי ולפני מימושו כאלגוריתם הוא דורש מחשבה יצירתית.

לעומת זאת, במקרים רבים, הבעיות שמתכנתים מתמודדים איתן במציאות הן מורכבות ויש בהן אלמנטים רבים של חוסר וודאות. לכן, מציאת הדרך לפתרון הבעיה אינה תמיד פשוטה ואינטואיטיבית, וכדי לפתרה צריך דרך שיטתית הכוללת גם ניתוח הבעיה (המייצגת מציאות כלשהי). למעשה, תהליך הפתרון של בעיה מורכבת דורש הפעלת כישורים ויכולות כמו יצירתיות, זיהוי וניתוח סוגיות עיקריות, זיהוי אובייקטים וקשרים בין האובייקטים, הכרות עם תבניות ומצבים דומים, יכולת בחינת אסטרטגיות שונות וקבלת החלטות, יכולת תכנון ומימוש פתרון, ולבסוף גם יכולת בחינה והערכה של הפתרון.

כדי לשפר את יכולת פתרון הבעיות של תלמידים, אנו ממליצים להשתמש בבעיות המוגדרות כחידות.

חידה היא סוג של בעיה, המוגדרת במילון כתעלומה, דבר שתוס, לא ברור שקשה להבין אותו ושמעוניינים לפענח אותו. הגדרה נוספת לחידה היא בעיה הדורשת חשיבה לוגית או מתמטית לפתרונה.

תהליך פתרון בעיות

Polya (1957) תיאר תהליך פתרון בעיות המורכב מארבעה שלבים:

1. הבנה וניתוח הבעיה – הוא השלב הראשון שמהווה תנאי הכרחי לפתרון של בעיה. שלב זה כולל בין היתר: זיהוי המידע הרלבנטי בבעיה, זיהוי האובייקטים וניתוח קשרים ביניהם וזיהוי המטרה או הפתרון המבוקש.
2. מציאת אסטרטגיה – כדי לפתור את הבעיה יש למצוא דרך או אסטרטגיה המתאימה לפתרונה מתוך מגוון אסטרטגיות קיימות שאת חלקן נציג בחלק השני של מאמר זה.
3. יישום האסטרטגיה – זהו השלב בו מתבצע הפתרון בפועל. כלומר, הפעלת האסטרטגיה שנבחרה על המידע שהוגדר ונותח מתיאור הבעיה. במדעי המחשב, בשלב זה אנו גם מתכננים מודל פורמלי המתאים למימוש הפתרון על המחשב ומיישמים אותו כתכנית מחשב. (במתמטיקה - שלב זה כולל ניסוח של הפתרון באמצעות משוואות מתמטיות ופתרון).
4. בחינה והערכה של הפתרון – השלב האחרון הוא אימות ובדיקה האם אכן הפתרון מתאים לבעיה.

למרות שביצוע השלבים בפתרון הבעיה הוא ליניארי, יתכנו גם קשרים נוספים בין השלבים השונים, שאינם בהכרח עוקבים זה לזה, כמודגם באיור 1. לדוגמה, שלב הערכה ובחינה צריך להתבצע לאחר כל שלב אחר ולא רק בסיום התהליך.

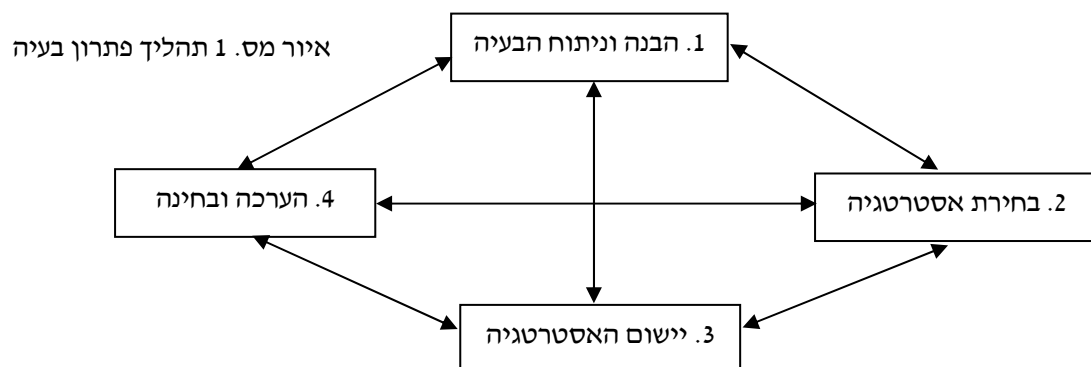
היתרון בשילוב חידות כחלק מהתנסות בפתרון בעיות הוא שחידות הן מטבען בעיות קצרות שלא דורשות ידע מוקדם להבנתן. ועם זאת, בדרך כלל פתרון מורכב יותר ומחייב שימוש במיומנויות לפתרון בעיות כמו: ניתוח זהיר של הבעיה, יצירתיות, בחירת אסטרטגיה ותכנון פתרון מתאים, ויכולת בקרה עצמית כדי לבדוק את נכונותו. כמו כן, פתרון חידות מהווה אתגר אינטלקטואלי המניע את הפותר לחתור לפתרונה, דבר הגורם סיפוק אישי ומשפר את המוטיבציה ללמידה. בניגוד לתפיסה הרווחת כי חידות הן בעיות המתאימות לתלמידים בעלי אוריינטציה מתמטית חזקה או תלמידים טובים במיוחד, אנו חושבים שניתן לשלבן במדעי המחשב כחלק מההוראה השוטפת לכלל התלמידים.

בהתבסס על אמונתנו בתועלת החינוכית של שימוש בחידות בהוראת מדעי המחשב, החלטנו השנה, ביחד עם מחשבה - מרכז המורים הארצי למדעי המחשב, להפעיל אתר ניסיוני המכיל תיאור של חידות ואת אופן הפעלתן בכיתה. החידות מתוארות באתר מחשבה, בפינה הקרויה **חידות ובעיות – ללמוד וללמד קצת אחרת**. החידות חולקו לנושאים שונים ובכל קובץ מתוארות החידות, הפתרון המלא שלהן והמלצות לפעילות בכיתה.

כתובת האתר

<http://cse.proj.ac.il/monthly/index.htm>

בחלק הראשון של מאמר זה נציג תהליך של פתרון בעיות. בחלק השני נתאר אסטרטגיות מקובלות לפתרון חידות ובחלק האחרון נתאר דוגמה לשימוש בחידה בהוראת יסודות 1.



אסטרטגיות לפתרון בעיות

אסטרטגיה היא שיטת פעולה או גישה כללית שמישהו מאמץ לצורך השגתה של מטרה מסוימת. בחירת אסטרטגיה מסוימת לפתרון בעיה דורשת הכרות עם מגוון אסטרטגיות קיימות והפעלת שיקול דעת בבחירת האסטרטגיה (או האסטרטגיות) המתאימה ביותר. כלומר, במציאת האסטרטגיה שלהערכתנו תביא לפתרון הבעיה. את האסטרטגיות ניתן לחלק לשני סוגים: אסטרטגיות כלליות, המתאימות לכלל הבעיות (אסטרטגיות 1 עד 6) ואסטרטגיות המתאימות לסוג מסוים של בעיות (אסטרטגיות 7 ו-8).

בחלק זה נציג מספר אסטרטגיות ואת אופן השימוש בהן לפתרון בעיות, כאשר חלק מן האסטרטגיות שנציג מוכרות לנו מפתרון בעיות אלגוריתמיות. שימוש באסטרטגיות אלו באופן שוטף, מאפשר לתלמיד להרחיב את רפרטואר האסטרטגיות לפתרון בעיות וכך לשפר את יכולתו בתחום זה.

1. **הפרד ומשול (Divide and Conquer):** זו אחת האסטרטגיות הכלליות שמקובלות מאוד בפתרון בעיות במדעי המחשב, והיא מבוססת על פירוק הבעיה לתת-בעיות ופתרון כל תת-בעיה בנפרד. בחלק הבא, המתאר שימוש בחידה בהוראת מדעי המחשב, השתמשנו גם באסטרטגיה זו.

2. **הקטנת הבעיה:** לפי אסטרטגיה ידועה זו, ההנחה היא שקל יותר למצוא פתרון עבור הבעיה המוקטנת ופתרון זה מתאים גם לפתרון של הבעיה המורכבת. לדוגמה, בבעיות קומבינטוריות כמו מציאת מספר האפשרויות לסידור n אנשים בטור, מתחילים בדרך כלל בסידור מספר קטן של אנשים (1,2,3) וכך הלאה) ולאחר מכן פותרים עבור n.

3. **אנלוגיה:** חיפוש תבנית פתרון מוכרת שניתן לאמץ או להתאים אותה גם בפתרון החידה החדשה. לדוגמה, כדי לסכם את סדרת המספרים הטבעיים מ-1 עד 100 ניתן לבצע את ההליך הבא: נחבר את המספר הראשון עם המספר האחרון (1+100), את המספר שלפני האחרון עם המספר השני (2+99) וכך הלאה.

קל לראות כי סכום כל שני מספרים הוא 101 וכי יש 50 זוגות שסוכמו. מכאן שסכום כל המספרים הוא 101×50 . נשתמש באסטרטגיה זו כדי לחשב את סכום הספרות בכל המספרים מ-1 עד 1,000,000. נסדר את זוגות המספרים הבאים: 0 – 999,999; 1 – 999,998; 2 – 999,997 וכך הלאה. סכום הספרות בכל זוג מספרים כזה הוא 54 ומספר הזוגות הוא 500,000 ולכן סכום הספרות בכל המספרים מ-1 עד 1,000,000 הוא $54 \times 500,000$. דוגמה נוספת הן החידות הדיופנטיות שפתרון כרוך בבניית מערכת משוואות, שבהן מספר הנעלמים גדול ממספר המשוואות ולכן יש אינסוף פתרונות. בפתרון חידות דיופנטיות אנו דורשים שהפתרון יהיה מספר שלם ותנאים נוספים הנובעים מאופי השאלה. בחלק השלישי של מאמר זה אנו משתמשים באנלוגיה זו.

4. **שימוש ברקורסיה:** על פי אסטרטגיה זו אנו מנסים למצוא מצבים פשוטים ולהתאים להם פתרון ולאחר מכן לבנות את הכלל איך להגיע ממצב מורכב יותר למצב פשוט ידוע. אחת הדוגמאות לשימוש באסטרטגיה זו (ואולי הנפוצה ביותר) היא בעיית מגדלי הנוי. דוגמה נוספת היא חידת חודש דצמבר שפתרונה הוא שימוש בסדרת פיבונאצ'י והיא מבוססת על שימוש באסטרטגיה זו.

5. **פתרון בגישת backward:** על פי אסטרטגיה זו אנו פותרים את הבעיה מהסוף להתחלה. כלומר, מתחילים מהתוצאה המבוקשת ומנסים למצוא את הדרך להגיע לנתוני הבעיה. לדוגמה, ברצוננו למצוא שיטת משחק המבטיחה נצחון במשחק בו יש ערימה של 50 גפרורים, כשכל שחקן בתורו צריך להסיר 1,2,3 או 4 גפרורים מהערימה. המנצח הוא השחקן שלקח את הגפרורים האחרונים. על פי גישת backward, קל יותר לבחון את המצב הסופי של המשחק בו נותרו: 1, 2, 3, 4 או יותר גפרורים ולנסות לבנות את הכללים למצב שלפניו, מאשר לבחון את הערימה ההתחלתית של הגפרורים ולתאר את כל המצבים האפשריים.

6. **חפש תכונה קבועה** (invariant): על פי שיטה זו אנו מנסים לזהות בבעיה מצב או תכונה שאינם משתנים גם אם נבצע פעולות שונות.

לדוגמה, בבעיה המתוארת באסטרטגיה השלישית (אנלוגיה), הסתמכנו על העובדה שסכום הזוגות בבעיה הראשונה וסכום הספרות בבעיה השנייה קבוע.

דוגמה נוספת: משחק עם חפיסת שוקולד המחולקת לריבועים ב-8 עמודות ו-5 שורות. כל שחקן בתורו מחלק את החפיסה לשניים (לאורך קווי הסימון בין הריבועים). לדוגמה, חלוקה ראשונה יכולה להיות לשני חלקים, הראשון 3×5 והשני 5×5 . המפסיד הוא השחקן שלא יכול לחלק את החפיסה (כלומר, כל החפיסה חולקה לחלקים שכל אחד מהם הוא ריבוע אחד). השאלה היא האם אנו יכולים לדעת מי ינצח במשחק? כדי לדעת מי מנצח נשתמש בעובדה שמספר ריבועי שוקולד הוא תמיד 40, כאשר בהתחלה יש חפיסה אחת בה 40 ריבועים, ובסיום יהיו 40 חפיסות בכל אחת מהן ריבוע אחד. כיוון שבכל שלב של המשחק מספר חפיסות השוקולד גדל ב-1 (אבל סך כל הריבועים נשאר קבוע), קל לראות כי אחרי 39 שלבים נקבל מצב סופי והמנצח הוא תמיד השחקן שביצע את החלוקה הראשונה.

7. **שימוש בעקרון שובך יונים**: זו אסטרטגיה המתאימה לסוג מסוים של בעיות על פי הכלל הבא: אם נשבץ יותר מ- n עצמים ב- n קופסאות, לפחות קופסה אחת תכיל לפחות שני עצמים. לדוגמה, אם יש קבוצה של 13 תלמידים לפחות, נוכל לטעון כי לפחות שניים מהם נולדו באותו חודש. נתאר דוגמה אחרת: ארזו 481 תפוזים ב-240 ארגזים. צריך להוכיח כי קיים (לפחות) ארגז אחד בו יש לפחות שלושה תפוזים. נציג הסבר אינטואיטיבי לפתרון: אם נרצה לשים 480 תפוזים ב-240 ארגזים כך שאף ארגז לא יכיל לפחות 3 תפוזים, נצטרך לשים 2 תפוזים בכל ארגז, ואז את התפוז הנוסף נצטרך לשים באחד מהם. כלומר, תמיד יהיה לפחות ארגז אחד שבו לפחות 3 תפוזים.

8. **שימוש בסימטריה**: במקרים מסוימים ניתן למצוא סימטריה בין מצבים או עצמים. לדוגמה, משחק מפורסם הוא משחק גפרורים שבו שתי ערימות וכל שחקן בתורו יכול להסיר מספר גפרורים מאחת הערימות. השחקן שהסיר את הגפרור האחרון הוא המנצח. במשחק זה, האסטרטגיה להבטיח ניצחון היא הסימטריה. כדי לנצח, שחקן צריך לדאוג לכך שלאחר מהלך שלו מספר הגפרורים בשתי הערימות יהיה שווה (סימטרי). לדוגמה: מתחילים בערימות שיש בהן 4 ו-7 גפרורים. האסטרטגיה המנצחת היא להביא למצב שבו בשתי הערימות אותו מספר. ולכן כדי לנצח, השחקן הראשון צריך להסיר 3 גפרורים מערימה של ה-7. כעת המצב הוא (4,4). בגלל הסימטריה, לכל פעולה שהשחקן השני יבצע, השחקן הראשון יכול לבצע אחריו פעולה שתגרום למצב בו מספר הגפרורים בשתי הערימות יהיה שווה, ולכן הוא יבצע את ההסרה האחרונה.

דוגמה לשימוש בחידה

החידה הבאה (מופיעה באתר מחשבה כחידת חודש ינואר), היא חידה עתיקה ומפורסמת. תחילה נתאר את החידה:

אישה הלכה לשוק למכור ביצים. בדרך לשוק, עבר רוכב על סוס ובשוגג פגע בסל הביצים והן נשברו.

הרוכב שרצה לפצות את האישה, שאל אותה: כמה ביצים היו לך בסל?

ענתה האישה: אני לא זוכרת בדיוק כמה ביצים שמתי בסל, אבל כשסידרתי את הסל והוצאתי מהסל בכל פעם 2 ביצים, נשארה רק ביצה אחת בסל. כך גם קרה כשהוצאתי בכל פעם 3 ביצים, 4 ביצים, 5 ביצים ו-6 ביצים. אבל כשהוצאתי בכל פעם 7 ביצים מהסל, לא נשארה אף ביצה בסל.

חשב הרוכב מספר רגעים ואז אמר: אני אתן לך X ביצים שהן המספר המינימלי שמקיים את התנאים שתיארת.

השאלה היא: מה מספר הביצים שהציע הרוכב לאישה? הפתרון הוא: מספר הביצים שהציע הרוכב הוא 301. פתרון מלא לבעיה ניתן למצוא באתר מחשבה בחידת חודש ינואר.

- בשלב השני נציע שיפור לאלגוריתם: כדי לייעל את התכנית, נשתמש במידע שקיים בחידה. לדוגמה, אנו יודעים כי number מתחלק ב-2 עם שארית 1. כלומר, המספרים שכדאי לבחון אותם הם אי-זוגיים ואז אפשר לכתוב את ההוראה לדילוג:

$number \leftarrow number + 2$

באופן דומה אנחנו יודעים שהמספר מתחלק ב-5 ולכן המספרים יכולים להסתיים ב-0 או ב-5 (0, 5, 10, 15, ...). למספרים אלו נוסיף 1 (כי ידוע שיש בחלוקה ב-5 שארית 1), וכך נקבל את המספרים (1, 6, 11, 16, ...). אבל כל המספרים המסתיימים ב-6 הם זוגיים וזה סותר את התנאי הקודם. לכן, ניתן לשפר ולקפוץ ב-10 בכל פעם, כלומר:

$number \leftarrow number + 10$

- שלב שלישי: כדי לשפר את האלגוריתמים שתוארו קודם לכן בצורה משמעותית, רצוי לחשב מהם המספרים שמתחלקים ב-2, 3, 4, 5, 6 ללא שארית ולהם נוסיף 1. לדוגמה, מספר מתאים הוא: $2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 = 720$.

אבל חישוב זה לא מבטיח שזה המספר הקטן ביותר שניתן למצוא, למשל גם 360 מתחלק ב-2, 3, 4, 5 ו-6. ולכן כדי למצוא את המספר הקטן ביותר, נחפש את המספר הקטן ביותר שפרוק לגורמיו מכיל את המחלקים בחידה.

למעשה התלמידים מכירים בעיה דומה והיא: מציאת הגורמים של מספר נתון. אולם הפעם הבעיה היא הפוכה. אנו יודעים את הגורמים של המספר ומחפשים את המספר עצמו. במספר שאנו מחפשים, הגורמים שצריכים להיות הם 2, 3, ו-5 (משום שהם מספרים ראשוניים), אבל $2 \cdot 3 \cdot 5 = 30$, והוא אינו מתחלק ב-4. מאחר ו-4 ניתן לייצג כ- $2 \cdot 2$, נוסיף גורם נוסף 2, כלומר ננסה $2 \cdot 2 \cdot 3 \cdot 5 = 60$ המתחלק ב-2, 3, 4, 5 ו-6 ללא שארית. כלומר, מצאנו את המספר הקטן ביותר שאלו מחלקיו.

אסטרטגיות הפתרון מבוססת על שימוש בשתי אסטרטגיות שתוארו בחלק הקודם:

א. אנלוגיה: חידה זו שייכת למשפחה של חידות דיופינטיות.

ב. הפרד ומשול: כדי לפתור את הבעיה, עלינו לבצע את המשימות הבאות:

- למצוא סדרת מספרים שמתחלקים ב-2, 3, 4, 5 ו-6 עם שארית 1.
- מתוך סדרת המספרים שמצאנו ב-א', יש למצוא את אלו שמתחלקים ב-7 ללא שארית.

פעילויות בכיתה עם החידה

החידה מתאימה לתלמידים ביסודות 1 וקשורה למושגים: mod, div, לולאות, ניתוח בעיה, יעילות. בנוסף לפתרון המתמטי, ניתן גם לפתור את החידה על-ידי סריקה של המצבים האפשריים וסינון המצבים המתאימים.

הצעה להפעלה בכיתה:

- בשלב הראשון נציג את האלגוריתם הפשוט ביותר, אלגוריתם 1, שסורק את כל המספרים האפשריים (החל מ-1) ומוצא את המספר הראשון שעונה לתנאי. בכל אחד מהשלבים עלינו לנמק שהפתרון מציג את המספר המינימלי האפשרי של ביצים, ועונה לתנאי הבעיה.

אלגוריתם 1

$1 \leftarrow number$ (מספר הביצים)

כל עוד

(number לא מתחלק ב-2, 3, 4, 5, 6 עם שארית 1)

או (number לא מתחלק ב-7) בצע

$number \leftarrow number + 1$

הצג number

לאחר בדיקת האלגוריתם, ניתן לקיים דיון שיתמקד במושגים ניתוח ויעילות של אלגוריתם, ולבקש מהתלמידים הצעות לאלגוריתם יעיל יותר.

כעת עלינו לבצע את המשימה השנייה :

מבין המספרים שהם $60n+1$ (הכפולות של $60+1$), יש למצוא את המספר הראשון שמתחלק ב-7 ללא שארית. אלגוריתם 2 סורק את המספרים (החל מהערך הקטן ביותר) העונים לבעיה הראשונה ומנסה להתאימם לבעיה השנייה. המספר הראשון שעונה לשני התנאים הוא המספר המינימלי של הביצים.

אלגוריתם 2

$number \leftarrow 61$

כל עוד ($number$ לא מתחלק ב-7 ללא שארית) בצע

$number \leftarrow number + 60$

הצג $number$

- לבסוף, ניתן להרחיב את הבעיה ולבקש מהתלמידים לכתוב תכנית המחשבת ומציגה את 10 המספרים הראשונים שעונים לתנאי החידה. כלומר, 721, 301,
על פי הפתרון שתואר בשלב הקודם, עלינו למצוא את 10 המספרים שמקיימים את המשוואה $(60n+1) \div 7 = 0$. כדי שמשוואה זו תתקיים, n צריך להיות כפולה של 7, ולכן ההפרש בין שני המספרים הוא 420. כלומר, עלינו ליצור את הסדרה החשבונית שמתחילה ב-301 וההפרש בין כל שני איברים הוא 420. אלגוריתם 3 הוא מימוש של פתרון זה.

אלגוריתם 3:

$number \leftarrow 61$

לכל מספר מ-1 עד 10 בצע

$number \leftarrow number + 420$

הצג $number$

לסיכום

אתם מוזמנים להיכנס לאתר חידת החודש ולפורום המלווה אותו. מטרת הפורום היא להציג את החידות ולאפשר למורים לשתף פעולה ולהחליף ידע, רשמים והתלבטויות וכך להתנסות בפתרון החידות וליצור את מאגר החומרים שיתאימו לכלל האוכלוסייה.

בהזדמנות זו, אנו רוצים להודות לכל המורים שהשתתפו בפורום מאז פתיחתו ובעיקר לאיתן ראט ומאיר פריש, על ההערות, העצות, והחידות ששלחו.

מקורות

1. Polya, G. How to solve it. (1957) Garden City, NY: Doubleday and Co., Inc.

כתובת האתר:

חידות ובעיות – ללמוד וללמד קצת אחרת

<http://cse.proj.ac.il/monthly/index.htm>

יום עיון "הוראת מדעי המחשב – לאן?"

ביום רביעי 9.6.03 נקיים במכון ויצמן יום עיון בנושא "הוראת מדעי המחשב לאן?"

יום העיון יוקדש לשלושה הבטים של הוראת מדעי המחשב:

מהות הדיסציפלינה של מדעי המחשב,

מעמד המקצוע במערכת החינוך בארץ,

שיווק המקצוע לתלמידים, הורים ומערכת החינוך.

פרטים נוספים והרשמה: באתר המרכז הארצי <http://cse.proj.ac.il>

פיצוחים: משחק לימודי בנושא מבנה המחשב

אהובה שפרלינג ודורית ליקרמן

תיכון ליאובק חיפה

כללי המשחק

משחק תחרותי בין שתי קבוצות.

קבוצה ראשונה צריכה לעבור על פני מסלול מימין לשמאל והקבוצה השניה מלמעלה למטה.

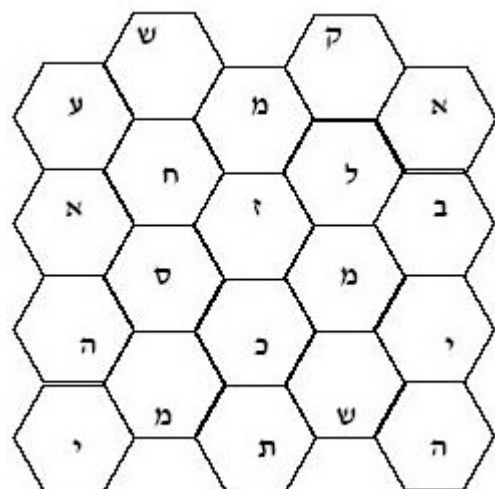
לכל קבוצה יש צבע משלה.

לאחר שמוגרלת אחת מהאותיות שעל הלוח, המורה קורא הגדרה למילה המתחילה באותה האות. המשבצת מסומנת בצבע של הקבוצה שגילתה את המילה המתאימה.

הקבוצה שגילתה את המילה המתאימה להגדרה בוחרת את האות הבאה וניתנת לה זכות ראשונים לענות על הגדרתה.

משבצת שההגדרה שלה לא נתגלתה לאחר 3 פעמים ברציפות – נחסמת. (שימו לב: האות ע נחסמת לאחר 2 הגדרות, האותיות ז, ח, ט נחסמות לאחר הגדרה אחת). בהגדרות המופיעות בהמשך, ליד כל אות מופיע מספר המספר מציין את כמות המשבצות המסומנות באותה האות. כל שלוש הגדרות מתאימות למשבצת (כי לאחר 3 הגדרות נחסמת המשבצת של אותה האות). מנצחת הקבוצה שהצליחה לסמן מסלול על הלוח כנדרש.

לוח המשחק



הגדרות

א (2)

עליו כותב המעבד את כתובת תא הזיכרון ממנו יש לטעון נתון מסוים, כתוצאה מכך הנתון נטען לאפיק הנתונים – **אפיק הכתובת** (Address Bus).

רכיב אלקטרוני המורכב ממספר מסוים של סיביות (כל סיבית היא הספרה 1 או 0) ומשמש לאחסנת מספר לשם ביצוע חישוב מסוים – **אוגר** (Register).

שפת סף המהווה ייצוג שמי של הוראות המכונה בשפה מובנת לעין האנושית – **אסמבלי**.

אליו נטען נתון שכתובתו נכתבה ע"י המעבד באפיק הכתובת – **אפיק הנתונים** (Data Bus) המעבד יכול לקרוא או לכתוב לזיכרון לכן אפיק הנתונים הוא דו כיווני.

סדרה של הוראות מדויקות לביצוע משימה נתונה – **אלגוריתם**.

שפת סף המהווה ייצוג שמי של הוראות המכונה בשפה מובנת לעין האנושית – **אסמבלי**.

ב (1)

סדרה של 8 סיביות מקובצות יחד – **בית** (Byte).

תפקידו להעביר את הנתונים מהמעבד להתקן קלט/פלט או מהתקן קלט/פלט למעבד או אל הזיכרון – **בקר**.

משימה שבה מתוארות נקודת מוצא ומטרה, ונדרש מתכנן להשגת המטרה – **בעיה אלגוריתמית**.

ה (2)

מילה ו/או רצף מילים שהמחשב "מבין" – **הוראה**.

זוהי הסיבית (Most Significant Bit) בקצה השמאלי של מספר בינארי. סיבית זו קובעת אם המספר חיובי או שלילי – **הסיבית המשמעותית ביותר**.

תהליך בו מתורגמת התכנית משפה עילית לתכנית בשפת מכונה – **הידור**.

זוהי הסיבית (List Significant Bit) בקצה הימני של מספר בינארי. סיבית זו קובעת אם המספר זוגי או אי זוגי – **הסיבית הכי פחות המשמעותית**.

תהליך בו מתורגמת התכנית משפה עילית לתכנית בשפת מכונה – **הידור**.

השלב בו מבצע המחשב את הוראות התכנית בשפת מכונה – **הרצה**.

ז (1)

בו נמצא קוד התכנית לביצוע, ערכי הנתונים ומידע נוסף, לכל תא בו יש כתובת ודרכה טוענים אותו למעבד – **זיכרון**.

ח (1)

אוסף רכיבים אלקטרוניים ומכניים הפועלים במשולב, יחד עם מעבד – **חומרה** (Hardware).

י (2)

היא המפעילה והמנהלת את מכלול החומרה שבמחשב – **יחידת העיבוד המרכזית**.

היא מבצעת את הפעולות החשבוניות (חיבור, חיסור, כפל וחילוק) ואת פעולות ההשוואה הלוגיות (קטן מ-, גדול מ-, שווה ל-) הנדרשות בביצוע תכנית מחשב – **יחידה אריטמטית לוגית** (ALU).

ביצוע זה קיימות 16 ספרות, המספרים מיוצגים על-ידי הספרות 0-9 והאותיות A,B,C,D,E,F – **ייצוג הקסה-דצימלי**.

מכוונת ומתזמנת את פעולת היחידות השונות במעבד ע"י שליחת/קבלת אותות אל/מהיחידות השונות – **יחידת בקרה** (Control Unit).

היא ה-CPU או בשמה המקוצר "המעבד" – **יחידת העיבוד המרכזית**.

ביצוע זה קיימות רק 2 ספרות וכל המספרים מיוצגים ע"י הספרות 0 ו-1 לכן נאמר שהן מיוצגות בבסיס 2 – **ייצוג מספרים בינארי**.

כ (1)

חלק חומרה, משמש לאחסון מידע – **כונן קשיח**.

התקן חומרה חיצוני לצריבת מידע **כונן תקליטורים**.

התקן חומרה חיצוני מאפשר לאחסן מידע נייד (כיום השימוש בו הולך ופוחת) **כונן תקליטורים**.

ל (1)

אחראי לתקשורת ולתזמון בין המעבד והיחידות האחרות המונחות עליו (כמו אמצעי קלט/פלט כרטיסים למיניהם רכיבי זיכרון פסי נתונים ופסי בקרה) – **לוח אם**.

מ (3)

מערכת אלקטרונית היכולה לקרוא נתונים ממקור קלט כלשהו, לעבד אותו ולהציג כפלט את תוצאת העיבוד – **מחשב**.

הוא המפעיל והמנהל את מכלול החומרה שבמחשב, ה-CPU או בשמו המלא "יחידת העיבוד המרכזית" – **מעבד**.

אמצעי קלט – **מקלדת**.

תוכנה המפעילה את ציוד המחשב ומספקת שירותים שונים למשתמש כגון: ניהול מערכת הקבצים, ניהול הזיכרון הראשי, הרצה וניהול תכניות ועוד – **מערכת ההפעלה**.

תכנית מחשב הקוראת תכנית הכתובה בשפה עילית מקובץ ומתרגמת אותה לשפת מכונה בעברית מהדר – קומפיילר (Compiler).

הוא המונח העברי ל file - קובץ.

ש (2)

היא מובנת למעבד מסוים שבלב המחשב – שפת מכונה.

היא מהווה ייצוג שמי של הוראות המכונה הבסיסיות בשפה מובנת לעין האנושית – שפת סף.

שפה זו פותחה בראשית שנות ה-70 ע"י דניס ריצ'י. היא מבוססת על השפות הקודמות B ו-BCPL ומכאן שמה – שפת C.

היא כתובה באופן טקסטואלי וכוללת הוראות נוחות לקריאת נתונים מהקלט, הגדרות משתנים בזיכרון, ביצוע פעולות חשבוניות ולוגיות, הדפסת נתונים לפלט וכו' – שפה עילית.

היא מהווה ייצוג שמי של הוראות המכונה הבסיסיות בשפה מובנת לעין האנושית – שפת סף.

שפה לכתובת יישומים שבהם נוח לייצג מידע באמצעות רשימות המבטאות קשרים בין נתונים. למשל Lisp או DrScheme - שפה פונקציונלית.

ת (1)

אוסף תכניות מחשב, כוללת בתוכה את מערכת ההפעלה ותכניות משתמש – תוכנה (Software).

הוא מנתח משימה המיועדת לביצוע במחשב, מפרק את המשימה לתת משימות פשוטות, כותב אותן מתכנון לביצוע ומיישמן ע"י תכנית מחשב – תכנת (Programmer).

מבנה נתונים מופשט הפועל על פי השיטה נכנס ראשון יוצא ראשון (First In First Out) - תור.

יחידה זו מאפשרת למעבד לבצע פעולת חיבור בין שני מספרים. באמצעותה ניתן לממש גם את פעולות החיסור והכפל – מסכם Adder.

התקן חומרה באמצעותו ניתן, למשל, לשלוח פקס ולהתחבר לרשת האינטרנט – מודם.

נמצא ביחידת הבקרה ואחראי לתרגום פקודת המכונה לרצף של אותות בקרה ליחידות השונות הנמצאות במעבד, המחוללים את ביצוע הפקודה בפועל – מפענח.

יחידת מידע בגודל שני בתים (16 סיביות) וכיום גודלה משתנה ממחשב למחשב. ישנם מחשבים בהם יחידה זו מכילה 4 בתים (32 סיביות) 8 בתים (64 סיביות) – מילה (word).

קוראת תכנית הכתובה בשפה עילית מקובץ ומתרגמת אותה לשפת מכונה – מהדר (Compiler).

ס (1)

יחידת המידע הבסיסית במחשב (ערכה יכול להיות 0 או 1) – סיבית (Bit).

התקן חיצוני בעזרתו ניתן, למשל, להעביר תמונה מנייר צילום למסך המחשב – סורק.

פס, עליו נהגו בעבר לכתוב הוראות בשפת מכונה וגם לאחסון נתונים – סרט.

ע (1)

על ידי נכתבת תכנית בשפה עילית לקובץ - עורך (Editor).

התקן קלט חיצוני – עכבר.

ק (1)

משמשים לתזמון ולבקרת רכיבי החומרה לצורך פעולה משולבת – קווי בקרה.

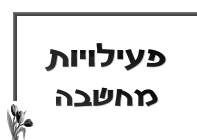
אתר האינטרנט של מרכז המורים הארצי למדעי המחשב

באתר האינטרנט של מרכז המורים הארצי למדעי המחשב נפתח לאחרונה קטלוג תכנים חדש המכיל קטגוריות שימושיות ונועד לסייע לכם להתמצא באתר ולגלוש ביתר קלות לפי תחומי העניין שלכם. הקטלוג הזה מתווסף לפורטל יחידות הלימוד שמאפשר לכם גישה לחומרים לפי יחידות הלימוד בתכנית הלימודים. הקישורים המופיעים בו יכוונו אתכם למאמרים ותקצירי מאמרים, מצגות, חומרים שפותחו על-ידי מורים, וקישורים רלבנטיים נוספים.

הקטגוריות הראשיות בקטלוג

תחומים במדעי המחשב: משתנים, פתרון בעיות, אלגוריתמיקה ועוד	שיטות הוראה במדעי המחשב: פרויקטים, מטפורות, ויזואליזציה ועוד	פעילויות למידה: פעילויות כיתה, פעילויות מעבדה, משחקים
מחקר חינוכי ותפיסות של תלמידים במדעי המחשב	הבטים רב-תחומיים של מדעי המחשב	הדיסציפלינה של מדעי המחשב: אישים והסטוריה

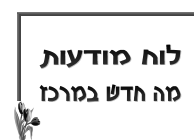
כדי לעודד אתכם להעזר בחומרים שנמצאים באתר, בחרנו בדוגמאות מהאתר במבוא למערכות מידע, וכן בשתי דוגמאות לחומרים שפותחו על-ידי מורים מובילים לסדנאות הפדגוגיות שהתקיימו השנה בכל הארץ. החומרים שפיתחו זיוה קונצמן ופריידה בן-פזי עומדים לרשותכם באתר יחד עם החומרים מהסדנאות האחרות ותוכלו למצוא אותם באתר הסדנאות <http://cse.proj.ac.il/workshops/index.htm>



פעילויות



הבטים



מה חדש?



יחידות לימוד



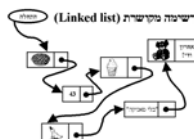
טופס הרשמה



קטלוג תכנים



תמורה



קישורים



פורומים

תורת המחשב: חומרים מסדנה פדגוגית של המרכז הארצי

זיוה קונצמן

תיכון הרצוג, כפר סבא

חלק א: אנליזה נומרית

שיטת החציה

שיטה דו-נקודתית, המחשבת בכל איטרציה את האמצע בין שתי נקודות, ומחליפה את אחת מהנקודות בנקודת האמצע, במידה והפונקציה בנקודה זו שוות סימן עם הפונקציה בנקודת האמצע.

חציה (דיוק, a,b)

{ האלגוריתם מקבל דיוק רצוי (יסומן כ-0.01 וכולי), קצוות קטע התחלתי (a,b) ויחזיר את הפתרון של פונקציה f כלשהי נתונה, בדיוק הנתון }

1. כל עוד $|a-b| < \text{דיוק בצע}$:

$$1.1 \quad \text{mid} \leftarrow (a+b)/2$$

1.2 אם $f(\text{mid}) * f(a) < 0$ אזי

$$1.2.1 \quad b \leftarrow \text{mid}$$

1.3 אחרת,

$$1.3.1 \quad a \leftarrow \text{mid}$$

2. החזר mid

באתר תוכלו למצוא תכנית לפתרון נומרי של משוואה בשיטת החציה עם קירוב נתון לערך המשתנה.

שיטת הסריקה הפרימיטיבית

שיטה המאפשרת איתור קטעים בהם קיים פתרון אחד לפחות של הפונקציה. מחלקים את הקטע החל מהנקודה הנתונה ב"קפיצות" שוות: גודל כל "קפיצה" נקבע לפי רמת הדיוק הנדרשת. אם הדיוק הוא 3 ספרות אחרי הנקודה, אזי אורך הקפיצה יהיה 10^{-4} . סורקים את הפונקציה בתחום הנבדק עפ"י הקפיצות שנקבעו, ואם סימן ערך הפונקציה בשלב מסוים שונה מסימן ערך הפונקציה בשלב הבא, הרי שמצאנו קטע מצומצם בו נמצא אחד משורשי הפונקציה.

סריקה פרימיטיבית (דיוק, a,b)

{ האלגוריתם מקבל דיוק רצוי (יסומן כ-0.01 וכולי), שתי נקודות שהן קצוות של קטע a,b וידפיס את כל הפתרונות של פונקציה f כלשהי נתונה, בדיוק הנתון }

$$1. \quad \text{fx1} \leftarrow f(a)$$

$$2. \quad \text{fx2} \leftarrow f(b)$$

3. כל עוד $a < b$ בצע:

$$3.1 \quad \text{אם } \text{fx1} * \text{fx2} < 0 \text{ או } \text{fx1} = 0$$

אזי הדפס a

$$3.2 \quad \text{fx1} \leftarrow \text{fx2}$$

$$3.3 \quad a \leftarrow a + \text{דיוק}$$

$$3.4 \quad \text{fx2} \leftarrow f(a)$$

באתר תוכלו למצוא תכנית רקורסיבית לפתרון נומרי של משוואה לא ליניארית בשיטה הפרימיטיבית.

שיטת ניוטון-רפסון

השיטה מסתמכת על כך, שבהינתן פונקציה, הניתנת להבעה ע"י נוסחה כלשהי, אפשר גם לרשום נוסחה עבור נגזרתה. אם זה כך, אפשר למצוא נוסחה למשיק של עקומת הפונקציה בנקודה נתונה, ואז בד"כ המשיק יהיה קירוב טוב לעקומת הפונקציה בסביבת הנקודה.

השיטה היא חד-נקודתית: מקבלים נקודה כלשהי x_0 , המהווה קירוב ראשון לשורש המשוואה. מעבירים בנקודה זו אנך לציר ה-x. בנקודה בה האנך חותך את הפונקציה, מעבירים משיק לגרף הפונקציה.

משיק זה חותך את ציר ה-x בנקודה x_1 , המהווה קירוב טוב יותר מאשר הנקודה הקודמת. כל שנשאר הוא לחשב את הנקודה השנייה, לפי ערך הנקודה הקודמת הנתונה וכך הלאה:

$$X_{n+1} = x_n - (f(x_n)/f'(x_n))$$

שיטת ניוטון-רפסון (דיוק, x_0)

והואיל ומתקיים $|\sin x| \leq 1$ לכל x , הרי ברור ש- y אינה יכולה להתאפס מחוץ לקטע $[-5, 5]$.

{האלגוריתם מקבל דיוק רצוי, נקודה התחלתית x_0 ויחזיר את הפתרון של פונקציה f כלשהי שתהיה נתונה בדיוק הנתון}

$$x_1 \leftarrow x_0 - (f(x_0)/f'(x_0)) \quad 1.$$

2. כל עוד $|x_0 - x_1| < \epsilon$ דיוק וגם $f(x_1) > 0$ בצע:

$$x_0 \leftarrow x_1 \quad 2.1$$

$$x_1 \leftarrow x_0 - (f(x_0)/f'(x_0)) \quad 2.2$$

3. החזר x_1

אלגוריתם למציאת שטח מתחת לעקומה בשיטת הטרפזים

שטח_למספר_טרפזים(a, b, n)

{האלגוריתם מקבל שני קצות קטע a, b ומספר טרפזים n ומחזיר את שטח n הטרפזים מתחת לפונקציה נתונה f }

$$h \leftarrow (b-a)/n \quad 1.$$

$$s \leftarrow f(a) + f(b) \quad 2.$$

$$c \leftarrow a \quad 3.$$

4. כל עוד $c \leq (b-h)$ בצע:

$$c \leftarrow c + h \quad 4.1$$

$$s \leftarrow s + 2 * f(c) \quad 4.2$$

$$s \leftarrow s * h / 2 \quad 5.$$

6. החזר s

שטח_לפי_טרפזים(דיוק, a, b)

{האלגוריתם מקבל שני קצות קטע a, b ודיוק ומחזיר את השטח מתחת לפונקציה נתונה בתחום זה בדיוק {הרצוי}

$$n \leftarrow 5 \quad 1.$$

$$s_2 \leftarrow \text{שטח_למספר_טרפזים}(a, b, n) \quad 2.$$

3. חזר

$$s_1 \leftarrow s_2 \quad 3.1$$

$$n \leftarrow n * 2 \quad 3.2$$

$$s_2 \leftarrow \text{שטח_למספר_טרפזים}(a, b, n) \quad 3.3$$

4. עד אשר $|s_2 - s_1| > \epsilon$ דיוק

5. החזר s_2

באתר המרכז הארצי תוכלו למצוא דוגמאות מפורטות נוספות לשאלות מבחינות הבגרות. החומרים נמצאים באתר תורת המחשב (הכניסה דרך פורטל יחידות לימוד).

מציאת קטע התחלתי

כאשר לא נתון קטע התחלתי, מותר להפעיל כל ידע מתמטי כדי להגיע בקירוב לקצות קטע.

נניח שברצוננו למצוא x המקיים:

$$2^x = 2 - x$$

כלומר, ברצוננו למצוא את השורשים של:

$$f(x) = 2^x + x - 2$$

ננסה לאתר נקודה או נקודות חיתוך של גרף הפונקציה הזו עם ציר ה- x . לשם כך נגדיר שתי פונקציות עזר:

$$u(x) = 2^x, v(x) = 2 - x$$

נשרטט את שני הגרפים על מערכת צירים אחת.

מהשרטוט רואים מייד ש- $u(x)$, שהיא פונקציה מעריכית, היא פונקציה עולה, ואילו $v(x)$, שהיא פונקציה קווית – יורדת. ולכן:

- שתי הפונקציות נחתכות בנקודה אחת
- נקודה זו היא בין 0 ל- 2, וזהו "בגדול" הקטע ההתחלתי.

דוגמה נוספת:

נניח שברצוננו לפתור את המשוואה

$$X^2/25 = \sin x$$

כלומר, עלינו לדון בפונקציה:

$$Y = X^2/25 - \sin x$$

כאשר ערכי x מבוטאים ברדיאנים, רואים ש:

$$X^2/25 > 1 \text{ עבור } |x| > 5$$

$$(5^2/25 = 1) \quad ((-5)^2/25 = 1)$$

חלק ב: תורת הגרפים

ריכוז מושגים

עץ פורש מינימלי: רק בגרף משוקלל, $n-1$ הקשתות עם המשקל הנמוך ביותר.

גרף קשיר היטב: גרף מכוון, שבו עבור כל זוג צמתים קיים מסלול מכוון.

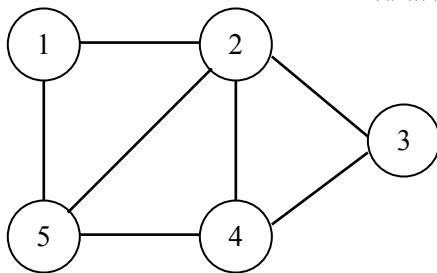
גרף תשתית: הגרף המתקבל בגרף מכוון, לאחר הורדת הכיוון בקשתות.

משפט אוילר בגרף מכוון: גרף מכוון הוא גרף אוילר, אם ורק אם גרף התשתית שלו קשיר, ומתקיים **אחד** משני התנאים הבאים:

בצומת ההתחלה מספר דרגות היציאה גדול ב-1 מדרגות הכניסה. עבור כל שאר הצמתים, דרגת הכניסה = דרגת היציאה, ועבור כל צומת, דרגת הכניסה = דרגת היציאה.

ייצוג על-ידי מטריצת סמיכויות

מטריצת הסמיכויות המייצגת גרף G היא מטריצה שממדיה $|V| \times |V|$. בתא i, j יהיה 1 אם קיימת קשת בין קודקוד i לקודקוד j או 0 במקרה ואין קשת כזאת. (ישנם גם ייצוגים בצורת רשימות של קשתות וצמתים, אך כאן נדגים רק בעזרת מטריצת הסמיכויות) דוגמה:



	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0

גרף: מבנה המורכב מקבוצת קודקודים (צמתים) וקבוצת קשתות כל קשת מחברת שני צמתים.

גרף לא מכוון: גרף שאין בו חשיבות לסדר הצמתים המתארים קשת (ללא חצים).

גרף מכוון: גרף בו יש סדר בין הצמתים המרכיבים קשת, יש כיוון לקשת (יש חצים).

גרף משוקלל: גרף בו לכל קשת יש משקל.

דרגה של צומת: מספר הקשתות הקשורות לצומת.

דרגת כניסה של צומת: מספר הקשתות שראשן בצומת.

דרגת יציאה של צומת: מספר הקשתות שזנבן בצומת.

צומת סמוך לצומת: אם יש ביניהם קשת.

חוג עצמי: קשת, ששני קצותיה נוגעים באותה צומת.

גרף קשיר: גרף לא מכוון שבו קיים מסלול בין כל שתי צמתים בו (ניתן להגיע מכל צומת לכל צומת).

גרף שלם (מלא): גרף שבין כל שני צמתים בו יש קשת.

מסלול: רצף של קשתות מצומת מקור לצומת יעד.

מעגל: מסלול שצומת ההתחלה וצומת הסיום זהים.

מסלול אוילר בגרף לא מכוון: מסלול הבנוי מסדרת קשתות המקיימות: כל קשת בגרף נמצאת במסלול, וכל קשת מופיעה במסלול פעם אחת בלבד.

גרף אוילר: גרף שיש לו מסלול אוילר.

מעגל אוילר: מסלול אוילר שהוא מעגל.

משפט אוילר בגרף לא מכוון: גרף הוא גרף אוילר אם מתקיים אחד מהתנאים הבאים:

בדיוק שני צמתים הם בעלי דרגה אי-זוגית וכל השאר הם בעלי דרגה זוגית,

וכל הצמתים הם בעלי דרגה זוגית – מעגל אוילר.

מסלול המילטון: מסלול העובר דרך כל הצמתים בגרף.

עץ פורש: קבוצת הקשתות שמכסה את כל צמתי הגרף, שחייב להיות קשיר וללא מעגלים.

חיפוש לעומק (DFS) Depth-First-Search

מעבר לעומק – (DFS) depth-first traversal

1. סמן את כל הצמתים בגרף כ- `unvisited`
2. עבור כל צומת v בגרף בצע :
 - 2.1 אם צומת v מסומנת כ- `unvisited` אזי בצע מעבר(v)

מעבר(v)

1. סמן את v כ- `visited`
2. בקר את v
3. עבור כל הצמתים w הסמוכים ל- v בצע :
 - 3.1 אם w מסומן כ- `unvisited` אזי בצע מעבר(w)

חיפוש לרוחב (BFS) breadth-first traversal

מעבר לרוחב (בעזרת תור)

(צומת S היא צומת המקור)

1. סמן את כל הצמתים כ- `unvisited`
2. סמן את צומת המקור s כ- `visited`
3. אתחל-תור $q \leftarrow s$
4. $k[s] \leftarrow 0$
5. הכנס-לתור(q, s)
6. כל עוד לא תור-ריק(q) בצע :
 - 6.1 הוצא-מתור(q) u
 - 6.2 קדם(k, u)

קדם(k, u)

1. עבור כל קשת היוצאת מ- u ל- w בצע :
 - 1.1 אם $k(w)$ היא `unvisited` בצע
 - 1.1.1 $k[w] \leftarrow k[u] + 1$
 - 1.1.2 הכנס-לתור(w, q)
 - 1.1.3 סמן את w כ- `visited`

מעבר לרוחב (ללא תור)

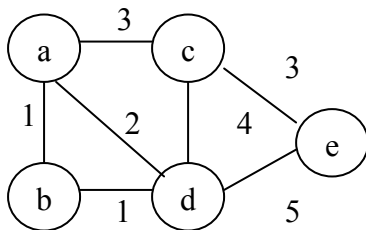
סמן את כל הצמתים כ- `unvisited`

1. סמן צומת S ב- 0
2. $I \leftarrow 0$
3. כל עוד צומת T אינה מסומנת בצע :
 - 3.1 סמן את כל הצמתים הסמוכים לצמתים המסומנים ב- I ב- $I+1$ וסמן אותן כ- `visited`
 - 3.2 הוסף 1 ל- I

באתר המרכז הארצי (אתר תורת המחשב) תוכלו למצוא גם תכניות מפורטות: מעבר לעומק, מעבר לרוחב, האם יש מסלול באורך K מצומת לצומת, אורך המסלול הקצר ביותר, הסגור הטרוניטיבי, האם_גרף_קשיד? (G) .

עץ פורש מינימלי

מחפשים את $n-1$ הקשתות שסה"כ יתנו את המשקל הזול ביותר ויכסו את כל קודקודי הגרף (שיש לו n קודקודים).



הגישה החמדנית: מחפשים מכל קודקוד שכבר יש, את הצלע הכי זולה (בתנאי שלא ייווצרו מעגלים).

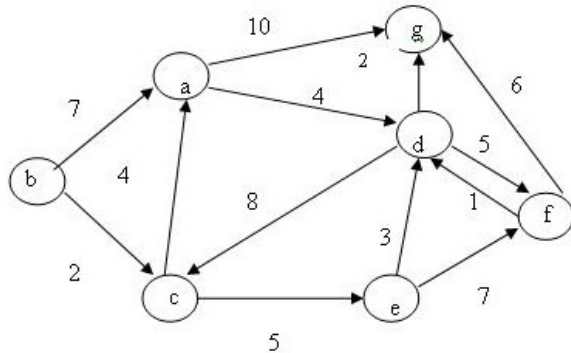
	V(1)	V(2)	V(3)	V(4)
הקודקוד	A	B	D	C
הקשת	(a,b)	(b,d)	(a,c)	(c,e)
המשקל	1	1	3	3

סה"כ המשקל = 8

בעמוד הבא תוכלו לראות את העץ המתקבל.

האלגוריתם של דייקסטרה: מציאת המסלול הזול ביותר
בגרף משוקלל מצומת S לצומת T

נתון הגרף הבא:



יש למצוא את המסלול הזול ביותר מצומת b לצומת g

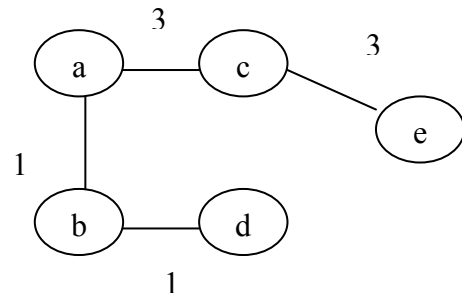
את כל השכנים של b מסמנים במרחק שלהם מ-b.
מבין כל השכנים של b בוחרים ומסמנים את הצומת שמרחקו הקטן ביותר – צומת c.
עכשיו ממשיכים עם הצומת המסומן באותה הדרך.
מבין השכנים של c הצומת הקטן ביותר הוא a.
מסמנים בכל צומת את המרחק המצטבר מצומת המקור. כלומר, מוסיפים את משקל הקשת האחרונה למשקל שרשום בצומת שממנה ממשיכים.
במידה ובצומת מסוימת יש כבר מספר, שהוא קטן יותר מהסכום החדש – לא משנים.
בכל שלב מסמנים את הצומת בעלת המספר הנמוך ביותר שטרם סומנה.
הגענו לצומת היעד g והמספר הוא 12. כלומר, משקל המסלול הזול ביותר מ-b לצומת g הוא 12 והמסלול עובר דרך:

$b(2), c(4), a(4), d(2), g$

אתר מומלץ להדמיית האלגוריתם של דייקסטרה
http://math.eitan.ac.il/graph_theory/200_Short/202_Short_Dij.htm

באתר המרכז הארצי (אתר תורת המחשב) תוכלו למצוא גם שאלות למבחני מתכונת וחומרים רבים נוספים.

קבלנו עץ פורש מינימלי:



עץ פורש מינימלי (E, V, T)

E – קבוצת קשתות הגרף

V – קבוצת קודקודי הגרף

T – קבוצת הקשתות שתהווה

את העץ הפורש המינימלי

1. $T \leftarrow \{ \}$ (קבוצה ריקה)

2. כל עוד (מספר הקשתות ב- T קטן ממספר הקודקודים

ב- V פחות 1 וגם E לא ריקה) בצע:

2.1 בחר קשת (u,v) מהקבוצה E בעלת משקל מינימלי

2.2 מחק את הקשת (u,v) מהקבוצה E

2.3 אם הוספת הקשת (u,v) אינה יוצרת מעגל,

הוסף אותה לקבוצה T

3. חזור T

שיטה נוספת: ממיינים את הקשתות בסדר עולה של המשקלים

(a,b)	(b,d)	(a,d)	(a,c)	(c,e)	(c,d)	(d,e)
1	1	2	3	3	4	5

מסיימים כאשר מגיעים ל- $n-1$ קשתות (ושוב, לא מכניסים קשת אם היא סוגרת מעגל).

$T = \{(a,b), (b,d), (a,c), (c,e)\}$

עץ פורש מינימלי

1. מיינ את כל הקשתות לפי סדר עולה של המשקלים.

2. בחר קשת ראשונה.

3. בדוק אם הצלע הבאה במשקל אינה סוגרת מעגל –

אם לא – צרף אותה לעץ

4. המשך כך, עד שתגיע ל- $n-1$ קשתות

חלק ג: אוטומטים ודקדוקים

כללים בסיסיים שימושיים:

4. יצירת תלות – כאשר המרכיבים אינם סמוכים זה לזה במלה – ההפקה תבצע תמיד יחד של אותם מרכיבים תלויים, וכאשר תסתיים הפקתם תוכנס ביניהם ההפקה הבלתי-תלויה.

דוגמה:

$$L = \{a^n b^m c^n \mid n, m \geq 0\}$$

$$S \rightarrow aSc / B, B \rightarrow bB / \varepsilon$$

1. הופעה אחת לפחות או מיד מלה ריקה – הפעלה חד-פעמית ראשונית ואחריה שליחה להפקה חוזרת זהה אך עם אפסילון.

דוגמה: הופעה אחת לפחות

$$S \rightarrow aA, A \rightarrow aA / \varepsilon$$

ללא הופעה אחת חובה:

$$S \rightarrow aS / \varepsilon$$

5. מלה ריקה – או לא

סיום עם האות או עם ε

דוגמה:

$$S \rightarrow aSb / a$$

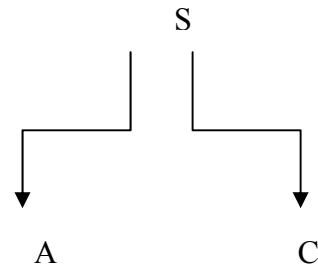
2. יצירת שני חלקים שונים לחלוטין בדקדוק, אחד אחרי השני, ללא כל תלות ביניהם – הפקת שתי אותיות גדולות מייד בהתחלה

דוגמה:

$$S \rightarrow AB, A \rightarrow aA / \varepsilon, B \rightarrow bB / \varepsilon$$

B, A יכולים להיות כל דבר מורכב המתפתח עצמאית.

דוגמה:



6. יצירת יחס של שונה

הפקתם יחד בכמות שווה ואח"כ יציאה לשתי אפשרויות שונות – הראשון גדול מהשני או ההיפך.

דוגמה:

$$L = \{a^n b^m \mid n < m, n, m > 0\}$$

$$S \rightarrow aSb / A / B$$

$$A \rightarrow aA / a$$

$$B \rightarrow bB / b$$

7. לימוד טכניקת אוטומט מכפלה, למקרה של תנאי וגם מורכבים.

$$L = \{a^n b^m c^m \mid n, m > 0\}$$

$$S \rightarrow AC$$

$$A \rightarrow aBb$$

$$B \rightarrow aBb / \varepsilon$$

$$C \rightarrow cC / c$$

8. יצירת מלים עם שארית חלוקה במספר כלשהו.

דוגמה:

$$L = \{a^i b^j c^k \mid i, j, k \geq 0\}$$

k השארית המתקבלת מחלוקת i ב-3

$$S \rightarrow aaaS / aBc / aaBcc / B$$

$$B \rightarrow bB / \varepsilon$$

3. יצירת תלות – להתחיל תמיד עם שוויון, ותמיד להפיק יחד (כמו מזרקה-כהגדרת שירלי)

$$S \rightarrow aSb / \varepsilon$$

דוגמה:

במידה ומדובר על תלות של גדול/קטן, לאחר הפקה בשוויון לאפשר רק הפקה של האות שנותרה.

דוגמה:

$$L = \{a^n b^m \mid n > m, m \geq 0\}$$

$$S \rightarrow aSb / A, A \rightarrow aA / a$$

9. יצירת ראי-

$$\Sigma(W) = \{b, c\}, L = \{aWaaW^R a\}$$

$$S \rightarrow aAa$$

$$A \rightarrow bAb / cAc / aa$$

10. יצירת מספר זוגי/אי-זוגי של מופעים

$$L = \{a^n b^m \mid n \bmod 2 = 0, m \bmod 2 = 1, n, m \geq 0\}$$

$$S \rightarrow aaS / aaA, A \rightarrow bbA / b$$

יצירה כל פעם של 2 מופעים, והסיום תלוי בדרישה:
אם הדרישה לזוגי אז מסיימים ב-2 מופעים, אם
הדרישה לאי-זוגי – מסיימים במופע אחד.
לא לשכוח, שזוגי זה גם 0 מופעים.

שגיאות אופייניות

מרבית השגיאות נובעות קודם כל מהבנת הנקרא.

- אם ההגדרה היא "מופיע" – הרי שהכוונה למקום כלשהו במלה ולא דווקא בהתחלה או בסוף.
- אם ההגדרה היא "אחרי כל b" או "כאשר יופיע b" – אין הכוונה שחובה שיופיע b.
- אם ההגדרה היא "יש במחרוזת 2 מופעים של b..." – אין הכוונה שכל המופעים של b יענו על התנאי, מספיק שיש 2 כאלו. הדבר גורר כמובן שגיאה אופיינית של יצירת מצבי מלכודת ואי-קבלת מלה שהיא בשפה.
- אם ההגדרה היא "2 מופעים עוקבים של a" – אין הכוונה שבהכרח הם יופיעו אחד אחרי השני, ביניהם יכולות להופיע אותיות אחרות.
- אם ההגדרה היא: "בין כל שני רצפים עוקבים של aים מופיעים aים ברצפים מהסוג a או aa" – השגיאה היא להניח שתמיד ה-aים יופיעו ברצפים כאלו, וכאילו חובה בכלל שיהיו aים במלה. מכאן נגזר כמובן שהמלה הריקה לא בשפה, למרות שהיא כן, או שמלה של a-ים בלבד אינה בשפה, למרות שהיא כן.
- אם ההגדרה היא "לפחות 2 a" – מגיעים למצב המקבל אחרי 2 a, ושוכחים לעשות "חוג עצמי" במצב המקבל עם a.
- זהירות בניסוח שאינו חד-משמעי
דוגמה: שפה המכילה מלים, בהן יופיע הרצף abb ולא תופיע האות c לפני רצף ה-abb הראשון. האם הכוונה ש-c לא יופיע כלל לפני הרצף הראשון של abb או בדיוק לפניו, בצמוד לו?

שגיאות שאינן קשורות להבנת הנקרא

- רישום אוטומט לא מלא – שוכחים לוודא שמכל מצב ייצא כל הא"ב של השפה.
 - שוכחים את המקרה הפרטי של המלה הריקה.
 - הכנסת אי-דטרמיניזם – מאותו מצב עם אותה האות מובילים לשני מצבים שונים.
 - שגיאה אופיינית – כאשר נדרש רצף ויוצרים דקדוק שבו אין רצף, עקב הפקה של גורמים תלויים. (באתר המרכז הארצי תוכלו למצוא דוגמה לשגיאה ספציפית של תלמיד).
 - שגיאה אופיינית באוטומט – חזרה למצב במהלך המלה, אשר מתאחד עם מצב שמאפיין חלק ראשון מיוחד של המלה. (באתר המרכז הארצי תוכלו למצוא דוגמה לשגיאה ספציפית של תלמיד).
 - שגיאה נפוצה – לא מציינים במצב מלכודת את כל הא"ב של השפה.
 - שוכחים לציין את המצב המקבל – כואב הלב ...
 - לא שמים לב כאשר באותה הפקה יש גם ε - ואז אינם שמים לב כתוצאה מכך, שאם יש שתי הפקות כל פעם, האחת אינה זהה בהכרח לשניה.
 - דוגמה: $S \rightarrow AbAb, A \rightarrow aaA / \varepsilon$ – תשובה שגויה נפוצה – מספר ה-aים זוגי.
 - סימון המצב ההתחלתי ע"י חץ כפול או מספור באוטומט או דקדוק המטפלים בחלוקה במספר מסויים, לא מתייחסים לכל השאריות האפשריות
 - שוכחים להתייחס למקרה של "שווה אפס" בהגדרת השפה.
 - אינם מתייחסים (או טועים) לקשר מספרי (כפול, מחצית וכולי) בין מרכיבים שונים בשפה.
- באתר המרכז הארצי (אתר תורת המחשב) תוכלו למצוא חומרים נוספים, כמו טכניקת מעבר מאוטומט לדקדוק (ולהיפך), דוגמאות לדקדוקים "בלתי-נסבלים" במיוחד, שאלות אפשריות למתכונת וחומרים נוספים.

<http://cse.proj.ac.il/torat/index.htm>

יסודות מדעי המחשב: חומרים מסדנה פדגוגית של המרכז הארצי

פריידה בן-פזי

תיכון הימלפרב, ירושלים

פירוק בעיה לתת-משימות

בפרק השלישי של מבחן הבגרות צריך לענות על שאלה של 20 נקודות (אחת מתוך שתיים). בשנים האחרונות נוהגים בפרק זה לתאר מצב מורכב, ולכתוב:

א. פתח אלגוריתם ש ...

- פרק את הבעיה לתת-משימות, כך שכל תת-משימה תיפתר באמצעות תת-תכנית. הגדר את המטרה של כל תת-משימה (הגדר טענת כניסה וטענת יציאה לכל תת-משימה).
- בחר במשתנים עיקריים, הגדר את טיפוסיהם, ותאר את תפקידיהם.
- ב. כתוב תכנית בפסקל ליישום האלגוריתם שפיתחת בסעיף א.

איך כדאי לגשת לשאלה כזאת?

- א. רצוי לקרוא את השאלה מספר פעמים, כדי להבין אותה כראוי. לפחות 2-3 פעמים.
- ב. בקריאה השלישית, להתחיל לכתוב מילולית בטיטה: מהם המרכיבים העיקריים של התכנית. בדרך כלל מדובר על 3-4 מרכיבים עיקריים.
- ג. עבור כל מרכיב עיקרי, צריך לקבוע:
 - * האם הוא פונקציה או פרוצדורה
 - * מהם הפרמטרים שהוא מקבל
 - * מה מוחזר לתכנית (או מודפס)
- ד. לכל פונקציה ופרוצדורה יש לכתוב את הכותרת, טענת כניסה ויציאה, ולהשאיר מקום להשלים אחר כך את גוף הפונקציה/פרוצדורה.
- ה. להכין רשימה של המשתנים העיקריים בתכנית, וליד כל אחד לציין מה הטיפוס שלו, ולכתוב מילולית מה התפקיד שלו.
- ו. להשלים את כל הפונקציות והפרוצדורות.
- ז. לכתוב את כל התכנית. בדרך כלל, התכנית תהיה די קצרה ותכלול בעיקר זימונים לפונקציות ופרוצדורות.

תרגיל:

א. פתח אלגוריתם שיקלוט 83 מספרים שלמים, כל אחד בתחום בין 1-100.

לגבי כל מספר שנקלט, האלגוריתם יבדוק אם קיימים שני מספרים שלמים וחיוביים שמכפלתם שווה למספר שנקלט, וההפרש ביניהם 3. אם קיימים שני מספרים כאלו, ינתן כתשובה המספר הקטן מביניהם, ואם לא, התשובה אפס. האלגוריתם יבדוק אם מבין כל התשובות התקבלו כל המספרים בין 1-10. אם כן, יודפס 'YES', ואחרת יודפס 'NO'.
הערה: הנח כי הקלט תקין.

- פרק את הבעיה לתת-משימות, כך שכל תת-משימה תיפתר באמצעות תת-תכנית. הגדר את המטרה של כל תת-משימה (הגדר טענת כניסה וטענת יציאה לכל תת-משימה).
- בחר במשתנים עיקריים, הגדר את טיפוסיהם, ותאר את תפקידיהם.
- ב. כתוב תכנית בפסקל ליישום האלגוריתם שפיתחת בסעיף א.

אוסף שאלות בנושא פירוק בעיה לתת-משימות

ההוראות מתייחסות לכל אחת מהשאלות בהמשך:

- א. בחר משתנים עיקריים, הגדר את טיפוסיהם ותאר את תפקידיהם.
- ב. בצע פירוק של הבעיה לתת-בעיות, כך שכל תת-בעיה תיפתר באמצעות פרוצדורה או פונקציה. ציין את הכותרת של כל פרוצדורה או פונקציה; כלומר, שם הפרוצדורה או הפונקציה, הפרמטרים שלה, וטענות כניסה ויציאה מפורטות.
- ג. כתוב תכנית בשפת פסקל לפתרון הבעיה על פי האלגוריתם שפיתחת (אין צורך לחזור על טענות הכניסה והיציאה שפרטת בסעיף ב').

שאלה 1

- ב. באיזה חודש/חודשים חל המספר הקטן ביותר של ימי הולדת?
- ג. באיזה חודש/חודשים יש יותר ימי הולדת לבנות מאשר לבנים?
- יש לפתח ולישם אלגוריתם שהקלט בו הוא נתונים על 200 חברי הקורס, והפלט שלו הוא הדפסת התשובות לשאלות הועדה.

שאלה 4

- לקראת כניסתה של חברת טלפונים סלולריים חדשה לשוק, ערך משרד התקשורת סקר לבדיקת שביעות רצונם של הלקוחות מהחברה.
- הסקר כלל 100 לקוחות מהחברה. כל לקוח קיבל מספר סידורי מ-1 עד 100. לטלפון הסלולרי של כל לקוח הוצמד חיישן שמנה את מספר הפעמים בחודש שהיתה הפרעה בתקשורת.
- יש לפתח ולישם אלגוריתם שהקלט שלו הוא 100 זוגות נתונים, זוג אחד עבור כל לקוח כאשר הנתון הראשון בכל זוג הוא מספר ההפרעות בחודש הראשון והנתון השני בכל זוג הוא מספר ההפרעות בחודש השני. פלט האלגוריתם יכלול שלושה מרכיבים:
- א. הודעה האם סה"כ ההפרעות (של כל הלקוחות) בחודש השני קטן מסה"כ ההפרעות (של כל הלקוחות) בחודש הראשון.
- ב. מספרי המנויים שממוצע ההפרעות שלהם גדול מממוצע ההפרעות של החברה בשני החודשים ביחד.
- ג. מספר המנוי שמספר ההפרעות שלו בחודש הראשון הוא הקרוב ביותר למספר ההפרעות הממוצע (לכל הלקוחות) בחודש הראשון. הנח שיש רק מנוי אחד כזה.

שאלה 5

במפעל נעלים ארבע מחלקות יצור:

- מחלקה 1 - נעלי נשים
- מחלקה 2 - נעלי גברים
- מחלקה 3 - נעלי ילדים
- מחלקה 4 - נעלי ספורט

- לכל סוג נעל קיים מספר בקטלוג המורכב מ-3 ספרות. ספרת המאות מציינת את קוד המחלקה (1-4) ושאר שתי הספרות מציינות את קוד הדגם. לדוגמה, המספר 354 בקטלוג מציינן נעל ילדים אשר קוד הדגם שלה הוא 54.

שחקן זוכה בהגרלת ה"לוטו" אם הצליח לנחש ששה מספרים שעלו בגורל מתוך 49 מספרים. השחקן מעוניין להעלות את סיכוייו לנצח. לשם כך פיתח אלגוריתם שיעלה את סיכוייו לנצח. השחקן קלט את נתוני תוצאות חמשת המחזוריים האחרונים והחליט שבמשחק הבא יבחר שישה מספרים אקראיים מתוך המספרים שלא עלו אף פעם בגורל בחמשת המחזוריים האחרונים. יש לפתח ולישם אלגוריתם שהקלט שלו הוא תוצאות ההגרלות של חמשת המחזוריים האחרונים, בכל מחזור ששה מספרים.

הפלט הוא: ששה מספרים אקראיים שונים מתוך המספרים שלא עלו בגורל.

שאלה 2

- כנס בינלאומי של החברה להגנת הטבע יימשך חמישה ימים. בתכנית הכנס מתוכננות שש סדנאות מדי יום בו-זמנית. לכנס מגיעות קבוצות ממדינות שונות. לצורך תכנון הסדנאות מארגני הכנס מעוניינים באיסוף המידע מהקבוצות השונות.
- קלט האלגוריתם הוא: שלשות מספרים המייצגים מספר סידורי של היום (1..5), מספר הסדנה (1..6) ומספר המשתתפים בסדנה. סדרת הקלט מסתיימת בזקיף של שלושה אפסים.
- פלט האלגוריתם הוא:
- א. המספר הסידורי של היום ומספר הסדנה, עבורה הביקוש הוא מקסימלי (הנח שיש רק סדנה אחת כזו).
- ב. הודעה האם ישנה סדנה ביום השלישי, עבורה אין כל ביקוש.

שאלה 3

על מנת לשפר את האקלים החברתי בקורס "כור היתוך" נקבעה וועדת תרבות. משימתה הראשונה של הוועדה היה לערוך רישום מדויק של כל חבריה. על כן נתבקש כל משתתף למסור את הפרטים הבאים: גילו (במספרים שלמים בין 14 ל-20), מינו (b/g) והחודש בו חל יום הולדתו (1-12).

כעת, הוועדה מבקשת לענות על השאלות הבאות:

- א. האם קבוצת הגיל 18 מאוזנת? קבוצת גיל (כל החברים שבאותו גיל) נקראת "מאוזנת" אם הפער בין מספרי הבנים והבנות בקבוצה אינו עולה על 5.

פתח ויישם אלגוריתם הקולט רשימה של מספרי קטלוג של נעלים אשר נמכרו בחודש האחרון. הרשימה מסתיימת במספר 0. הנח כי הקלט תקין. על האלגוריתם להדפיס את מספר זוגות הנעלים שמכרה כל מחלקה במשך החודש ואת מספר זוגות הנעלים שמכר המפעל. עליך להשתמש בפונקציה Hundred המחזירה את ספרת המאות של מספר חיובי שלם בן 3 ספרות.

שאלה 6

בכתה ט' בבית ספר "תהילה" 40 תלמידים. לכל תלמיד מספר סידורי מ-1 עד 40 המייצג אותו. המורה לדרמה המלמד בכתה הנ"ל בחר תלמידים לשתי הצגות: הצגת "הקוסם מארץ עוץ" והצגת "כנר על הגג".

להלן דוגמה לשתי קבוצות השחקנים.

הקוסם מארץ עוץ:

40 7 1 13 19 28 34 14 5 8 4 2

כנר על הגג:

17 27 33 32 25 11 21 10 5 9 18 4 3 2

מספר אמהות פנו למחנך הכתה בטענה שישנם תלמידים שנבחרו לשתי ההצגות בעוד שילדיהן לא נבחרו אפילו להצגה אחת. מחנך הכתה פנה אליך בבקשה לבדוק:

- כמה תלמידים נבחרו לשתי ההצגות ומהם מספריהם הסידוריים?
- מהם מספריהם הסידוריים של התלמידים שלא נבחרו אפילו להצגה אחת?
- כמה מתלמידי הכתה נבחרו בדיוק לאחת משתי ההצגות (או "הקוסם מארץ עוץ" או "כנר על הגג") ומהם מספריהם הסידוריים?

הקלט הוא שתי רשימות של מספרים סידוריים. רשימה ראשונה כוללת את מספריהם הסידוריים של המשתתפים ב"הקוסם מארץ עוץ" והרשימה השנייה כוללת את מספריהם הסידוריים של המשתתפים ב"כנר על הגג". כל אחת מהרשימות מסתיימת בזקף 0.

פרוצדורות

מתי משתמשים?

- כאשר רוצים להפריד חישוב מסוים או פעולה מסוימת לתת-תכנית, על מנת לקרוא לתת-תכנית בעת הצורך.
1. כדי לחלק את התכנית לחלקים קצרים ומובנים יותר.
 2. כדי לבצע הדפסות.
 3. לחשב ערך או ערכים ולהחזירם לתכנית.

היתרונות

- התכנית קריאה יותר כשהיא מחולקת לחלקים קטנים.
- תיתכן אפשרות להשתמש בקטע הפרוצדורה מספר פעמים בלי לכתוב את כל הקטע שוב ושוב.

כותרת הגדרה

1. אפשרות פשוטה: בלי לשלוח פרמטרים
שם procedure;
2. לשלוח פרמטרים ולהדפיס תוצאה בתוך הפרוצדורה
(פרמטרים נשלחים) שם procedure;
3. לשלוח פרמטרים ולהחזיר את הערכים המחושבים לתכנית הראשית
(פרמטרים נשלחים) שם procedure;
- (פרמטרים מוחזרים) var;

זימון פרוצדורה (בתכנית)

זימון פרוצדורה הוא תמיד פקודה בפני עצמה (שורה נפרדת), שאינה משולבת בתוך פקודות אחרות.

דוגמה לפרוצדורה מסוג 2:

הבעיה: כתוב פרוצדורה המקבלת 3 ערכים ומדפיסה את ממוצעם.

כתוב תכנית הקולטת גבהים של שלושה אחים ומשקלים שלהם, ומדפיסה את הגובה הממוצע ואת המשקל הממוצע.

דוגמה לפרוצדורה מסוג 3:

הבעיה: כתוב פרוצדורה המקבלת 3 ערכים ומחזירה את ממוצעם.

כתוב תכנית הקולטת גובה ומשקל של כל אחד משלושה אחים, ומדפיסה את הגובה הממוצע ואת המשקל הממוצע.

פרמטרים משתנים

יש שני סוגים של פרמטרים משתנים (שכותבים לפניהם **var** בכותרת של הפרוצדורה):

- 1) פרמטר שנשלח לפרוצדורה עם **ערך מסוים**, אבל הפרוצדורה משנה את הערך ומחזירה אותו לתכנית עם ערך חדש. (עין בדוגמה procedure mayen בהמשך).
- 2) פרמטר שנשלח **ריק** לפרוצדורה, והפרוצדורה יוצרת עבורו ערך לראשונה ומחזירה אותו לתכנית. (עין בדוגמה procedure ribuim בהמשך).

שליחת מערך כפרמטר :

בכותרת של פונקציה ופרוצדורה, במקום בו מציינים את הטיפוס של כל אחד מהפרמטרים, אסור בפסקל לכתוב שם מורכב. לכן, כאשר שולחים **מערך כפרמטר**, מצהירים תחילה על **שם מקוצר** לטיפוס המערך בפקודת **type**.
דוגמאות:

```
type a10=array [1..10] of integer;
procedure maxmin(a:a10; var max,min:integer);
    {טענת כניסה : מערך a מטיפוס a10}
    {טענת יציאה : המקסימום של המערך יוחזר ב- max
    והמינימום בmin}
```

```
procedure ribuim(a:a10; var b:a10);
    {טענת כניסה : מערך a מטיפוס a10}
    {טענת יציאה : יוחזר מערך חדש b המכיל
    ריבועי איברי המערך a בהתאמה}
```

```
procedure mayen (var a:a10);
    {טענת כניסה : מערך a מטיפוס a10}
    {טענת יציאה : יוחזר המערך a לאחר מיונו
    לפי סדר עולה}
```

בדיקה בוליאנית במערך

האם כל איברי המערך מקיימים תנאי מסוים, או לא סוג א'

דוגמה : נתון מערך A של 20 מחרוזות.
האם כל איברי המערך מכילים 6 אותיות לפחות?

דרך א' - הדרך הפשוטה

```
arukim:=true;
for i:=1 to 20 do
    if length(A[i])< 6
    then arukim:=false;
if arukim=true
    then writeln
        ('כל המחרוזות מכילות 6 תווים לפחות');
    else writeln
        ('לא כל המחרוזות מכילות 6 תווים לפחות');
```

דרך ב' - הדרך היעילה עם **while**

```
arukim:=true;
i:=1;
while (i<=20) and (arukim=true) do
begin
    if length(A[i])< 6
    then arukim:=false
    else i:=i+1;
end;
if arukim=true
    then writeln
        ('כל המחרוזות מכילות 6 תווים לפחות')
    else writeln
        ('לא כל המחרוזות מכילות 6 תווים לפחות');
```

הערות כלליות לבגרות

- בשאלות המבקשות כתיבת תכנית שלמה, יש להקפיד לכתוב תיעוד -
 - * מה הקלט של התכנית,
 - * מה הפלט של התכנית,
 - * ומפתח מה מייצג כל משתנה.
- לא לשכוח לאפס מונים וסכומים.
- לקרוא כל שאלה לפחות פעמיים, ולזהות מה מבקשים:
 - * תכנית שלמה
 - * קטע של תכנית
 - * דוגמאות מייצגות של קלט
 - * טבלת מעקב
 - * מציאת שגיאה, ותיקונה
- לזכור לענות לפי מה ששואלים, ובסוף השאלה לבדוק האם ענית על כל החלקים.
- לבדוק שענית על כל השאלות שצריך.

באתר המרכז הארצי (אתר יסודות) תוכלו למצוא חומרים רבים נוספים מהסדנה של פריידה בן פזי.

<http://cse.proj.ac.il/yesodot/index.htm>



דרך ג' - הדרך המתוחכמת

```
i:=1;
while (i<=20) and (length(A[i])>=6) do i:=i+1;
if i>20 then writeln
  ('כל המחרוזות במערך מכילות 6 תווים לפחות')
else writeln
  ('לא כל המחרוזות מכילות 6 תווים לפחות');
```

האם ערך מסוים מופיע במערך, או כלל לא

סוג ב'

דוגמה: נתון מערך A של 20 מחרוזות.

האם מופיע הערך 'כהן'?

דרך א' - הדרך הפשוטה

```
nimza:=false;
for i:=1 to 20 do
  if A[i] = 'כהן' then nimza:=true;
if nimza=true then writeln('יש כהן')
else writeln('אין כהן');
```

דרך ב' - הדרך היעילה עם

```
nimza:=false; i:=1;
while (i<=20) and (nimza=false) do
  begin
    if A[i] = 'כהן' then nimza:=true
    else i:=i+1;
  end;
if nimza=true then writeln('יש כהן')
else writeln('אין כהן');
```

דרך ג' - הדרך המתוחכמת

```
i:=1;
while (i<=20) and (A[i] <> 'כהן') do i:=i+1;
if i<=20 then writeln('יש כהן')
else writeln('אין כהן');
```

ניתוח מערכות מידע: הצעות לארועים ופרויקטים

מורים מובילים בהנחייתם של שרה פולק וצביקה פירסט

אחד המודולים בקורס מורים מובילים תשס"ד התמקד במקצוע "מבוא למערכות מידע" (שהיא אחת החלופות ביחידה השלישית במדעי המחשב). המורים המובילים, בהנחייתם של שרה פולק וצביקה פירסט, פיתחו חומרים מתאימים למקצוע זה. את החומרים שפותחו ניתן לסווג בצורה הבאה:

- א. הצעות לארגון למידה באמצעות פרויקט מתפתח (מהווה גרעין שמתפתח וניתן לשלבו לאורך השנה)
- ב. הצעה לפרויקט ותיאור פונקציונלי של מערכת המידע (דוגמה לתיאור פונקציונלי בתיק פרויקט).
- ג. אירועים שמתמקדים בתיאור פונקציונלי של מערכת מידע (מתאים לפרק 5).
- ד. אירועים שמתמקדים בתכנון ובניית מסד נתונים מנורמל (מתאים לפרק 4).

בחרנו להציג כאן מספר דוגמאות מכל סוג כדי להמחיש לכם את מגוון החומרים. באתר המרכז הארצי תוכלו למצוא פרוט נוסף, תרגילים ופתרונות. כתובת האתר <http://cse.proj.ac.il/db/index.htm>

1א. הצעה לארוע מתפתח: ספריה

עינת רוזנר

האירוע נבנה בהדרגה, תוך שאילת **שאלות מכוונות** בכל שלב. [הערות דידקטיות יופיעו בסוגריים]

בספריה העירונית משאילים ספרים לקוראים השונים שיש להם מנוי שנתי בספריה. בספריה יש ספרים בנושאים שונים. לכל ספר יכולים להיות 3 סוגי השאלות, לפי מספר ימי ההשאלה האפשריים: יום אחד, שבועה ימים, 30 יום. אחת לשבוע שולחת הספרינה תזכורות למנויים שלא החזירו ספרים לספריה 3 ימים אחרי מועד ההחזרה. בשלב זה, הספרינה רוצה לרכז את כל המידע על הספרים שיש בספריה.

שלב א

[הבאת האירוע בפני התלמידים.

שימוש במושגים: מסד נתונים, טבלה, יישות, תכונה, ערכים, מפתח ראשי, סכמה לטבלה.]

שאלה לתלמידים: כדי לנהל את המידע בספרייה אנחנו צריכים קודם כל מידע על ספרים. איזה מידע הייתם רוצים לדעת על כל אחד מהספרים?

פתרון אפשרי: שם ספר, שם סופר, שנת הוצאה, מספר ימי השאלה לספר.

[הערות דידקטיות:

במקרה שהתלמידים יוסיפו לטבלה שם תלמיד ששאל את הספר, נסיף גם תאריך השאלה ותאריך החזרה. רצוי לעורר דיון לגבי מאפיינים שאינם רלבנטיים לספריה (סביר להניח שמאפיינים כאלה יועלו ע"י התלמידים ואין להתעלם מהם). במידה ותתעורר שאלה לגבי העותקים של הספר, נסיף בשלב זה שדה עם מספר עותקים. יש לתת דוגמאות של כ- 3 ספרים שונים. מכאן ניתן לעבור להגדרת המושג **יישות, תכונות** של יישות **וערכים** של תכונות, תוך שימוש בדוגמאות שניתנו.]

בשלב זה ניתן לעבור **לכללי תקינות** על ערכי טבלה, ולהסביר על טיפוס הנתונים של כל אחת מהעמודות:

שאלות לתלמידים:

- האם שנת ההוצאה יכולה להיות גדולה מהשנה הנוכחית?
- מה המקסימום ומה המינימום של מספר ימי ההשאלה לספר, על סמך האירוע?

[הערות דידקטיות :

ניתן לשאול שאלות נוספות ולהסביר את כללי התקנות תוך הגדרת תחום הערכים לכל תכונה.

בשלב זה ניתן להסביר מהו מפתח ראשי. (הסבר אינטואיטיבי).

ניתן לקיים דיון על הגדרת אחת מהתכונות כמפתח, ולהוביל אותם לכך שיש צורך בהוספת שדה עם קוד ספר עבור המפתח.

לסיכום: הגדרת המושג סכמה של טבלה ורישום הסכמה לטבלת ספרים בעזרת התלמידים.]

שלב ב'

[פעולת מיון על קבוצת ישויות, ובניית קבוצת ישויות חדשה המקושרת לקבוצה זו.

בשלב זה רצוי שיקלטו מספר רשומות בתוך הטבלה, כדי שניתן יהיה להציג אותן בפני התלמידים. אם השיעור לא מתקיים במעבדת המחשבים, אפשר להדפיס את נתוני הטבלה.

כל השאלות נשאלות תוך כדי הדיון בכיתה.]

א. שאלה: נניח שהיתה לנו רשימה מאד ארוכה של ספרים והיינו רוצים לבדוק בצורה מרוכזת את הספרים של כל סופר. מה היה כדאי לנו לעשות עם הטבלה? מדוע כדאי למיין את טבלת הספרים?

[דיון על חשיבות המיון. רצוי להביא כדוגמה את רשימת האנשים הרשומים בספר טלפונים. הגדרת המושג שדה מיון. הגדרת המושגים סדר עולה וסדר יורד.]

ב. שאלה: נוסיף עוד 3 ספרים של הסופרת עירית לינור ("שתי שלגיות", "שירת הסירנה", "בנות בראון"). כיצד תהיה הטבלה מסודרת כעת?

ג. שאלה: נניח שבזמן הזנת שם הסופר, נקלטה האות הראשונה של השם ב- א' במקום ב- ע'. כיצד היו מוצגות הרשומות עכשיו?

ד. שאלה: האם היה ניתן למנוע את הקשת הטעות של שם הסופר ע"י כללי התקנות שנלמדו קודם?

ה. שאלה: נניח שאנחנו צריכים לדעת על הסופר, פרט לשמו, גם את שם הספר שהוא רב המכר של הסופר.

[שאלה זו מיועדת להביא אותם לצורך בטבלה נוספת שתשמור את הפרטים אודות הסופרים.]

ו. שאלה: איזה תכונות צריך לשמור על הסופרים? תשובה אפשרית: שם סופר, שם רב מכר.

ז. שאלה: מי מבין התכונות יכולה לשמש כמפתח? [שוב – ניתן לפתח דיון כמו בשלב א, שם הוספנו קוד ספר. נוסיף כעת מספר סופר בתור מפתח.]

ח. שאלה: איזה מידע על הסופר נצטרך לרשום בטבלת הספרים?

יש להוביל את התלמידים לכך שמספר הסופר יהיה השדה אותו נשמור בטבלת ספרים.

[בשלב זה יש להציג בפני התלמידים את טבלת הספרים המעודכנת ואת טבלת הספרים.]

ט. שאלה: איך נוכל לדעת מה שמו של הסופר שכתב את הספר "שירת הסירנה"?

[נצטרך להוביל את התלמידים לכך שנצטרך לבצע את הפעולות הבאות :

1. למצוא את הספר "שירת הסירנה" בטבלת הספרים, בעמודה שם ספר.

2. לבדוק בשורה זו את מספר הסופר שכתב את הספר.

3. לחפש בטבלת הסופרים בעמודה של מספר סופר, את השורה בה רשום מספר סופר זה.

4. נבדוק מה הערך המופיע בעמודה שם סופר בשורה זו בטבלת הסופרים (במקרה זה השם יהיה עירית לינור).

מכאן יש להוביל את התלמידים למושג מפתח זר ולקשר בין טבלאות.]

א.2. הצעה לארוע מתפתח: ספריה

שרה פולק

האירוע נבנה בהדרגה, תוך שאילת שאלות מכוונות בכל שלב. [הערות דידקטיות יופיעו בסוגריים]

[את אירוע הספריה נפתח בשלבים כאשר השלב הראשון מתייחס להגדרת הצרכים של הקורא ואחר הגדרת הצרכים של הספריה (או הספרינית). המטרה היא להדגים לתלמידים כי לכל מערכת מידע יש מטרות וצרכים ועל סמך זה נבנה מסד הנתונים.

הערה למורה: במידת הצורך ניתן להחליף נושא זה בספריית וידאו.]

שלב א: הגדרת המידע המאפיין ספר על פי צרכי השואל

[הערות דידקטיות :

שלב ראשון יוצא מנקודת מבט של התלמיד כלקוח השואל ספר בספריה. בשלב זה התלמיד יתאר מתוך ההתנסות שלו אילו סוגי שאלות הוא שואל את הספרנית ולאיזה סוג מידע בדרך כלל הוא מתייחס כשהוא נמצא בספריה. הרציונל הוא שלתלמיד קל יותר להגדיר את צרכיו כמשתמש, צרכים אלו הם בעצם חלק מיעדי מערכת המידע. לעומת זאת הגדרת צרכים מנקודת מבט של הספריה (או הספרנית) הם יותר ידע זר לתלמידים וקשה להם להתחבר אתו. בנוסף, תהליך זה יסייע לתלמידים בבואם להגדיר את מערכת המידע שלהם בביצוע הפרויקט.]

שאלה לתלמידים: כתלמידים המשתמשים בספריה, נסו להגדיר אילו שאלות אתם שואלים את הספרנית? איזה סוג מידע אתם מחפשים בספריה? כדי לנהל את המידע בספריה אנחנו צריכים קודם כל מידע על ספרים. איזה מידע הייתם רוצים לדעת על כל אחד מהספרים?

[נפריד את המידע והשאלות שתלמידים יעלו לשני סוגים :

מידע על הספר: השם, הסופר, הנושא, ...

מידע על המצאות הספר והאפשרות לשאול אותו: האם נמצא, היכן הספר נמצא, לכמה זמן מותר לקחת אותו (אם בכלל), ...

בשלב זה רצוי לעורר דיון לגבי מאפיינים שאינם רלבנטיים לספריה (סביר להניח שמאפיינים כאלה יועלו ע"י התלמידים ואין להתעלם מהם).]

שלב ב: בניית מסד נתונים לספר

[הערות דידקטיות: מטרת שלב זה לבנות מודל טבלאי לספר תוך שימוש במושגים: מסד נתונים, טבלה, יישות, תכונה, ערכים, מפתח ראשי, סכמה לטבלה.

מתוך השאלות שעלו בשלב הקודם נתייחס רק לשאלות הקשורות למידע על ספר ונגדיר מידע זה במטרה לבנות מסד נתונים שיכיל נתונים של ספר.

סיומו של שלב זה מאפשר תרגול במעבדה במסגרתו התלמידים אכן בונים את טבלת הספרים ומנסים לענות על השאלות שניסחו בשלב הקודם.]

פעילויות מוצעות :

- הקמת טבלה אחת

1. בהתאם לשאלות שניסחו בשלב הקודם, הגדרת תכונות של ספר ובשלב זה רק תכונות לא מרובות.

2. דיון והגדרת הטיפוסים של כל תכונה (בשלב זה לא חייבים להתייחס לעותקים של ספר, אך אם עולה השאלה נגדיר שדה מספר עותקים).

3. בחירת מפתח ראשי.

4. בניית טבלת ספרים ובצוע שאילתות.

- שימוש בטבלת עזר. לדוגמה, עבור קודים של נושאים.

- דיון בתכונה עם ריבוי ערכים: לדוגמה, מה קורה בספר שיש בו הרבה נושאים? כיצד מטפלים בארגון מידע זה בטבלה – קשר M:1.

[שאלה רלוונטית נוספת: מה קורה אם לספר יש מספר מחברים? כאן ישנה אפשרות לקבל מספר תשובות, כאשר אחת מהן גם מביאה לקשר רבים לרבים]

שלב ג: הגדרת הצרכים והמידע הקשור להשאלת ספרים

[הערות דידקטיות: נושא זה צריך להיות מתואר הן מהיבט של המשתמש והן מהיבט של הספרייה. המשתמש יתעניין בשאלות על מיקום ספר והאפשרויות לשאול אותו. הספרנית תהיה מעוניינת במידע שיסייע לה לעקוב אחר ספר: מי השאיל אותו, עד מתי, היכן נמצא הספר וכולי.

חלק מהשאלות הן משותפות הן לשואל והן לספרנית אך יתכן מידע שיעניין רק את הספרנית.

אוסף שאלות יגדיר את היעדים של תהליך השאלה (והחזרת ספר.]

פעילויות מוצעות :

לחזור על השאלות שהעלו התלמידים בשלב הראשון בו הגדירו את השאלות שהם שואלים בספריה, לאפשר לתלמידים להוסיף עוד שאלות.

המורה יתאר את התפקיד של הספרנית. מומלץ לבקש מהספרנית להגיע לכיתה, לאפשר לתלמידים לשאול אותה שאלות ולאפשר לספרנית לספר על הצרכים שלה כמשתמשת במסד המידע).

ב. הצעה לפרויקט ותאור פונקציונלי של מערכת

המידע: חברת בנייה

ויקטוריה צורי

הצגת הבעיה:

בני הוא מהנדס בנין שעובד בחברת בנייה ידועה באזור מודיעין. הוא אחראי על תיקונים שנעשים על ידי החברה בדירות שכבר נמסרו ללקוחות. עד היום הוא רשם את כל פניות של הלקוחות על דפים, ביומן או בפנקסים שונים, כך שחלק מהפניות נשכחו או טופלו מאוחר מדי. בעקבות כך הגיעו תלונות של דיירים רבים. מחוסר ברירה ולנוחיותו האישית החליט בני למחשב את הטיפול בלקוחות.

בני יצטרך לתחזק מאגר של לקוחות, בהם הוא יצטרך לטפל. הלקוחות ירשמו לפי הכתובת שלהם. זהו בעצם הנתון העיקרי שמעניין את בני. ניתן להוסיף לקוחות נוספים למאגר, אך אין לאפשר מחיקתם מהמאגר, עקב כך שדירות כבר נבנו ויש לשמור את נתוני הבעיות שהיו בהן.

כל אחד מהלקוחות יכול לדווח על בעיה כלשהי בדירתו. על בני לקלוט את הפניה, את סוג התקלה ולקבוע עם הדייר תאריך מתאים לביצוע התיקון. יתכן ובעיה מסוימת יצטרך לטפל יותר מיום אחד. כמו כן, הדייר יכול לפנות לבני בגלל אותה הבעיה יותר מפעם אחת.

כמו כן, עליו לשבץ צוות פועלים מסוים לצורך ביצוע התיקון. לכן יצטרך לשמור נתונים על הפועלים שלו ולשייך כל פועל לצוות מסוים. השיבוץ של הפועלים נעשה בהנחה שכל צוות יכול לטפל ביום אחד ביותר מבעיה אחת, אך בתאריך ובשעה מסוימים רק צוות אחד מטפל בבעיה נתונה. כל צוות מוכשר לטפל בתקלות מסוגים שונים. לכל צוות ישנו את הציוד הנדרש לצורך ביצוע התיקונים, ולבני אין צורך לטפל בזה.

ארגון: פרטי שישמש את בני בלבד.

מטרת על של המסד:

המטרה העיקרית של המסד שבני רוצה להקים היא לאפשר טיפול יעיל בפניות של הדיירים.

המטרות של המערכת:

- ניהול מידע על התקלות שיש אצל הדיירים.
- שיבוץ עובדים לביצוע התיקונים.
- ניהול מעקב אחר ביצוע התיקונים.

תוצאה של פעילות זו היא המידע והפעולות הדרושות לביצוע השאלה והחזרה. בנוסף, פעילות זו צריכה להדגיש את הצורך בעדכון של טבלאות: מי מטפל בקובץ ספרים שהקמנו בשלב קודם, איך מוסיפים ספרים חדשים וכדומה.

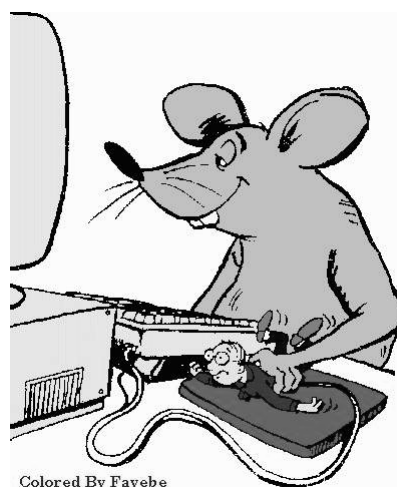
שלב ד: טיפול במידע הקשור להשאלת ספרים

[הערות דידקטיות: שלב זה הוא תכנון מסד נתונים שיאפשר לבצע את השאלת הספרים. בשלב זה נתייחס לקשרים בין טבלאות ופעולות לאחזור מידע משתי טבלאות. בנוסף נתייחס לצורך בעדכון טבלאות ובדיקות תקינות.]

פעילויות מוצעות:

נממש פעולות אלו בהדרגה:

1. לכל תלמיד מותר לשאול רק ספר אחד בכל פעם. מכאן שדרושה עוד טבלת קוראים שבה רשומים פרטי הספר שמושאל ויש קשר $M:1$ (לספר יש מספר עותקים שיכולים להיות מושאלים למספר קוראים).
2. לכל תלמיד מותר לשאול מספר ספרים בכל פעם – יוצר קשר $M:N$.
3. אפשרות לשאילת המשלבות שני מאגרים.
4. ביצוע פעולות עדכון על טבלאות.



היעדים:

1. תחום המערכת הנבדקת היא הפעילויות של בני, פעילויות הקשורות לביצוע תיקונים בדירות שנמסרו ללקוחות.
2. ישויות חוץ:
- א. חברת הבניה - המוסרת פרטים של דירות שנמסרו ללקוחות ופרטי לקוחות.
- ב. לקוחות - המוסרים פרטים על בעיות וליקויים ומבררים פרטים.
- ג. עובדים - המקבלים פרטים על תיקונים, מבצעים את התיקונים ומדווחים על ביצוע.
3. מטרות ויעדים לדוגמה:
- א. שיבוץ עובדים לביצוע התיקונים.
- קביעת תאריך וצוות לטיפול בבעיית הלקוח.
- הוספת עובד חדש לרשימת העובדים.
- שיבוץ העובד לצוות לפי התמחותו.
- שיבוץ צוות מתאים ופנוי לטיפול בתקלה.
- העברת תקלות שכבר טופלו לארכיון.
- הפקת מידע עבור כל התקלות שהיו בעבר בדירה נתונה.
- הפקת שמם של הפועלים שטיפלו בבעיה מסוימת בתאריך מסוים.
- פיטורין של עובד תוך כדי שמירה על תיקונים שהוא ביצע בארכיון.

יעדים במערכת מידע	תיאור האירוע: פעילות הקשורה לדיירים – ניהול מידע על תקלות שיש אצל דיירים.
הוספת פרטי דייר עדכון פרטי דייר אחזור פרטי דייר	להוסיף דיירים נוספים למאגר.
קליטת דיווח על בעיה עדכון פרטי בעיה שיבוץ צוות לתיקון	כל אחד מהדיירים יכול לדווח על בעיה כלשהי בדירתו. על בני לקלוט את הפניה, את סוג התקלה ולקבוע עם הדייר תאריך מתאים לביצוע התיקון.
אחזור פרטים על מצב בעיה	לקוח מברר על מצב טיפול בבעיה.

יעדים במערכת מידע	תיאור האירוע: פעילות הקשורה לעובדים – שיבוץ עובדים לביצוע תיקונים.
הוספת עובד חדש עדכון פרטי עובד אחזור פרטי עובד	לשמור נתונים על הפועלים שלו.
שיבוץ עובד לצוות על פי התמחותו	לשייך כל פועל לצוות מסוים. כל צוות מוכשר לטפל בתקלות מסוגים שונים.
שיבוץ צוות לביצוע תקלה	לשבץ צוות פועלים מסוים לצורך ביצוע התיקון.
עדכון דיווח על ביצוע תיקון	
ביטול נתוני עובד	פיטורין של עובד תוך כדי שמירה על תיקונים שהוא ביצע בארכיון.

ג. דוגמאות לאירועים שמתמקדים בתיאור פונקציונלי של מערכת מידע

ג1. הצעה לארוע: מיון לנבחרת כדורסל

מלכה סלמונוביץ

נבחרת הכדורסל העירונית של ראשון לציון מבצעת מידי עונה "גילוי כישרונות חדשים" לנבחרת. הנבחרת מונה בהרכב מלא: 10 שחקנים בבוגרים, 12 שחקנים בנוער, 15 שחקנים בילדים וכן מאמן, עוזר מאמן, מנהל קבוצה ומדריך כושר.

השחקנים הטובים ביותר עולים לשחק בחמישייה הפותחת במשחקים החשובים ביותר ושאר השחקנים מתחלפים בהתאם לקבוצות היריבות.

לקראת מבצע צירוף שחקנים לנבחרת מפרסמים אילו שחקנים (בוגרים, נוער או ילדים) דרושים העונה.

שחקן המעוניין להצטרף לנבחרת ממלא "טופס בקשה" בו פרטיו האישיים (שם, ת.ז, כתובת, טלפון, תאריך לידה) ומצרף אליו את רשימת הקבוצות הקודמות בהן שיחק ובאילו שנים. בנוסף, מצרף השחקן את פירוט עברו הרפואי הכולל רשימת מחלות כרוניות, ניתוחים קודמים ואישור רופא המצהיר כי השחקן כשיר לשחק כיום.

את טופס הבקשה מוסר הנרשם לפקידת הנבחרת. הפקידה מתייקת את כל הפניות בתיק "בקשות" ומעבירה אותן לעיונו של המאמן הראשי, הקובע אם יש מקום להמשיך ולבחון את בקשתו של כל פונה להיות שחקן בנבחרת.

לאור המלצות המאמן שולחת הפקידה לפונים מכתב דחייה ("אין באפשרותנו לקבלך כחבר בנבחרת") או זימון לבדיקות רפואיות.

כשמגיעות תוצאות הבדיקות הרפואיות (כשיר/לא כשיר) של הנרשמים פותחת הפקידה תיק לכל נרשם ומצרפת את התוצאות לתיק.

תיקי השחקנים המועמדים נמסרים לוועדת הקבלה של הנבחרת למיון. הוועדה לוקחת בחשבון את גילו של המועמד, מצבו הרפואי וניסיונו הקודם וממליצה על קבלתו או דחייתו. החלטות הוועדה נרשמות בתיק המועמד.

הפקידה שולחת לשחקנים המועמדים מכתבי דחייה או קבלה לנבחרת בהתאם להחלטת הוועדה.

ג2. הצעה לארוע: תיקון מכשירי טלפון נייד

גרוסמן רימה ופוק אלה, עריכה: עינת רוזנר

במעבדה לתיקון מכשירים טלפוניים ניידים מקבלים מכשירים לתיקון.

לקוח, המבקש לקבל שירות, מוסר את פרטיו (ת.ז, שם פרטי ושם משפחה, כתובת, טלפון), ופרטי המכשיר (מספר דגם, יצרן) וגם מתאר את התקלה. פקידה בקבלה בודקת האם הלקוח רשום במאגר הלקוחות, ובמידה והוא אינו מופיע, מוסיפה את פרטיו למאגר.

לאחר מכן, היא בודקת במאגר מכשירים ויצרנים האם המעבדה מטפלת במכשירים של אותו יצרן ובדגם המסוים שהלקוח מחזיק. במקרה ולא, המכשיר לא מתקבל לתיקון במעבדה.

אחרת, הפקידה רושמת את סוג התקלה בליווי טופס המכיל את פרטי הלקוח, סוג המכשיר ותאור התקלה.

בסיום התיקון מוסר הטכנאי את המכשיר בצירוף טופס אחריות ובו מפרט התקלה ופרטי התיקון. הטופס מתויק בקלסר תקלות והעותק נמסר ללקוח.

ד. אירועים שמתמקדים בתכנון ובניית מסד נתונים מנורמל

ד1. הצעה לארוע: מערכת בית משפט

שרית בכר ואבלסי עמליה

לצורך ניהול מערכת המשפט נתבקשת לטפל בתיקים הבאים:

נידונים (ת_ז נידון, שם משפחה, שם פרטי, כתובת, עיר, טלפון, תאריך לידה, סטטוס (חדש/וותיק))

שופטים (קוד שופט, סוג שופט (שלום/מחוזי/עליון), וותק)

עורך דין (קוד עו"ד, שם, התמחות (אזרחית / תעבורה / פלילי), שנת קבלת רשיון, שכר בסיס)

תיק בימ"ש (מספר תיק, תאריך פתיחת התיק, תאריך סגירה, האם נגזז(ל/כ), סיבת גניזת תיק (מחמת הספק / מחוסר ראיות / חוסר עניין))

הנחות יסוד :

1. שופט מטפל בתיקים רבים. תיק מטופל על-ידי שופטים רבים (בתאריך מסוים).
2. עורך דין מטפל בתיקים רבים. תיק מטופל על-ידי עו"ד אחד.
3. תיק בימ"ש כולל עבירות שונות. בתאריכים שונים נעשו עבירות שונות.

משימות :

1. על פי הנחות היסוד יש ליצור קשרי גומלין בין הטבלאות הכוללים בין היתר (במידת הצורך) הוספת טבלאות, הוספת שדות, הוספת מפתחות ראשיים / זרים.
2. היכן תציב את שדה תאריך טיפול שופט מסוים בתיק?
3. היכן תציב את שדה קוד עבירה? מה שם הטבלה שבה יופיע השדה הזה? והאם הוא יופיע בשתי טבלאות?

ד.2 הצעה לארוע: חברת השמירה

עינת רוזנר

- חברת שמירה "השומרים" מנהלת רישום של השומרים העובדים אצלה וחלוקתם לחוליות.
- כל שומר שייך לחולייה אחת וכל חולייה אחראית על איזור אחד או יותר. ייתכן כי יותר מחולייה אחת תהיה אחראית על איזור אחד.
- לכל שומר יכול להיות אחד מהתפקידים הבאים:
- אחראי חולייה, סגן אחראי חולייה, שומר רגיל.
- במקרה של אירוע, מוזעק אל האתר בו התרחש האירוע, אחד השומרים מאחת החוליות האחראיות על האיזור בו קרה האירוע, ומדווח על הממצאים.
- החברה מנהלת רישום של יומן אירועים.

משימות :

1. רשום את הנחות היסוד.
2. מהן היישויות המשתתפות בתהליך ומה המאפיינים (הנחוצים) של כל אחת מהן.
3. זהה את המפתח הראשי של כל יישות, ובנה רשימת קשרים (ציון את מידת הריבוי של הקשרים).
4. שרטט תרשים יישויות קשרים (ERD).

ד.3 הצעה לארוע: רשת האופנה קסטרו

שרית בכר ואבלסי עמליה

רשת האופנה קסטרו מעוניינת שניצור עבורה מסד נתונים המתייחס לפנ הייצור.

בחברה עובדים מספר רב של מעצבים לכל מעצב מתחזקת החברה את הנתונים הבאים: תז מעצב, שם משפחה, שם פרטי, טלפון, תאריך תחילת עבודה, התמחות.

המעצבים מעצבים בעבור החברה דגמים שונים לקולקציות שונות.

לכל דגם מתחזקת החברה את הפרטים הבאים: קוד דגם, שם דגם, עלות, צבע, מידה, כמות מינימום וכמות במלאי.

לכל קולקציה יש קוד המיצג את הקולקציה באופן חד ערכי, שם לקולקציה, תאריך תחילת קולקציה, תאריך גמר קולקציה, מבצע(כ/ל), אחוז הנחה.

כל דגם יכול להיות מורכב מסוג בד אחד וממספר רב של פריטים (כפתורים, רוכסנים ...)

בתכנון כל דגם בוחר המעצב את הבד המתאים לדגם ובנוסף בוחר הוא את הפריטים לדגם.

לכל פריט מתחזקת החברה את הנתונים הבאים: קוד פריט, שם פריט, כמות מינימום וכמות במלאי.

לכל בד מתחזקת החברה את הנתונים הבאים: קוד בד, סוג בד, ארץ יצור, רוחב בד, צבע, עלות בד למטר, כמות מינימום וכמות במלאי, הוראות יצרן.

להלן הנחות היסוד :

- מעצב יכול לעצב מספר רב של דגמים אך דגם שייך למעצב אחד.
- כל דגם נתפר מסוג בד מסוים אך מבד זה נתפרים מספר רב של דגמים.
- לדגם מספר רב של פריטים השייכים רק לו
- כל דגם שייך לקולקציה אחת בלבד אך בקולקציה דגמים רבים

(א) עזרו למנתח המערכות לזהות את הישויות לצורך בניית מודל ישויות הקשרים.

(ב) הוסיפו או מחקו תכונות בהתאם לנדרש

(ג) במידה והתווספו מפתחות זרים רשום מהם

(ד) הדגימו את הקשרים בין הטבלאות

תבל - מחשב ומחשבה

סביבת תבל היא סביבת למידה ממוחשבת המיועדת להכרות עם מושגי מפתח במדעי המחשב ועם מיומנויות חשיבה בסיסיות בפתרון בעיות תכנותיות. הסביבה וחומרי הלימוד הנלווים לה פותחו בקבוצת "מגוון" – מחקר ופיתוח בהוראת מדעי המחשב, הפועלת בטכניון מזה חמש עשרה שנה לקידום תכניות חינוכיות במדעי המחשב בבתי הספר התיכוניים.



משרד החינוך התרבות והספורט
4023 אישור מס'

LCSI של חברת MicroWorlds 2.0 בתוכנת

הקנדית, חברה בעלת שם עולמי ביישומי טכנולוגיה בחינוך.

חומרי הלימוד קיבלו לאחרונה את אישור משרד החינוך, התרבות והספורט.

סביבת הלמידה מאורגנת באופן מודולרי בחמש תקופות לימוד העוסקות בנושאים הבאים: (1) הכרת המחשב והסביבה הממוחשבת (2) תכנות אנימציה בעזרת הליכים ולולאות (3) תכנות מבני (4) משתנים (5) אלגוריתמים בסיסיים בתכנות. סביבת תבל וחומרי הלימוד הנלווים לה מותאמים לתכנית הלימודים הרשמית במקצוע מדעי המחשב בחטיבה העליונה. הלמידה בסביבה מבוססת על חקירה מודרכת במעבדת המחשבים ועל פיתוח פרויקטים תכנותיים שבמהלכם יוצרים ומפעילים אובייקטים מתחומי מדיה שונים (ראו טבלה).

תלמידי החטיבה העליונה יכולים לגשת בתום שנת לימודים אחת לבחינת בגרות בהיקף של 1 יח"ל במדעי המחשב (סמל הבחינה 899122); בחטיבות הביניים, ניתן להשתמש בפרקים נבחרים מתוך חומרי הלימוד במסגרת שיעורי מבוא לתכנות, הכרת המחשב ושיעורי העשרה.

אובייקטים הניתנים לתכנות בסביבת תבל (רשימה חלקית)

שם האובייקט	הכלי ליצירת האובייקט	תיבת העריכה	טיפוסי ערכים אותם ניתן להקנות לאובייקט שנוצר
צב			שם (פס עריכה) הוראה לביצוע – לא הכרחי (פס עריכה) בורר (radio button): בין ביצוע חד פעמי או רב פעמי של הוראה
עורך טקסט			שם (פס עריכה) תיבת סימון להצגת שם, תיבת סימון להסתרת מסגרת (transparent), תיבת סימון להצגת האובייקט (visible)
עורך מנגינה			שם (פס עריכה) תיבת סימון להצגת שם, תיבת סימון להצגת האובייקט (visible) בורר כלי הנגינה, בורר מקצב התווים קלידים, שנאי עוצמת קול, שנאי קצב כפתור Play, כפתור stop
כפתור			שם (נתון מראש) הוראה לביצוע (פס עריכה) בורר (radio button): בין ביצוע חד פעמי או רב פעמי של הוראה
שנאי ערכים			שם (פס עריכה), תיבת סימון להצגת שם ערך מינימלי (פס עריכה) ערך מקסימלי (פס עריכה)



סביבת תבל משווקת לבתי הספר במסגרת ערכה שעלותה 200 ש"ח ושומרכיביה הם:

- שתי חוברות עבודה לתלמיד המכילות הנחיות לעבודה עצמית בשיעורי המעבדה ודפי סיכום ומשימות נוספות (סימ"ן).
- לומדה תומכת עם קבצים מיוחדים בהם משתמשים התלמידים בשיעורי מעבדה (גירסת תבל 2.0).
- ספר מדריך למורה.



לתשומת לבכם – יש לרכוש בנפרד רישיון שימוש עבור MicroWorlds 2.0 של חברת LCSI. ניתן לרכוש רישיונות במחיר מיוחד למוסדות חינוך בישראל ע"י פנייה ל"מגוון" 8292880 – 04. פרטים נוספים באתר תבל - cse.proj.ac.il/tevel/index.htm



הבטים בהוראת מדעי המחשב: משוב לגליון יוני 2004

קוראים יקרים

המרכז הישראלי לחינוך מדעי טכנולוגי (מל"מ) מתנה את המשך המימון של כתב העת בקבלת משוב מהקוראים ובראיות שיש די קוראים המעוניינים בעיתון.
אם ברצונכם להמשיך לקבל את "הבטים בהוראת מדעי המחשב", אנא מלאו ושלחו משוב זה בהקדם למטה מל"מ.
תודה על שיתוף הפעולה,
צוות המרכז הארצי

משוב לגליון יוני 2004 של "הבטים בהוראת מדעי המחשב"

נא למלא ולשלוח אל:

מטה מל"מ, מגדל השמש

מכון ויצמן למדע

ת.ד. 26 רחובות 76100

פקס 08-9344162

1. שם בית הספר _____
2. שם המשיב _____ מספר המורים שעיינו בגליון זה בבי"ס _____
3. סמנו במשבצת המתאימה את חוות דעתכם:

הערות

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

4. הערות נוספות:
