



החוג למדעי המחשב  
המרכז האקדמי לחינוך  
מכללת בית-ברל



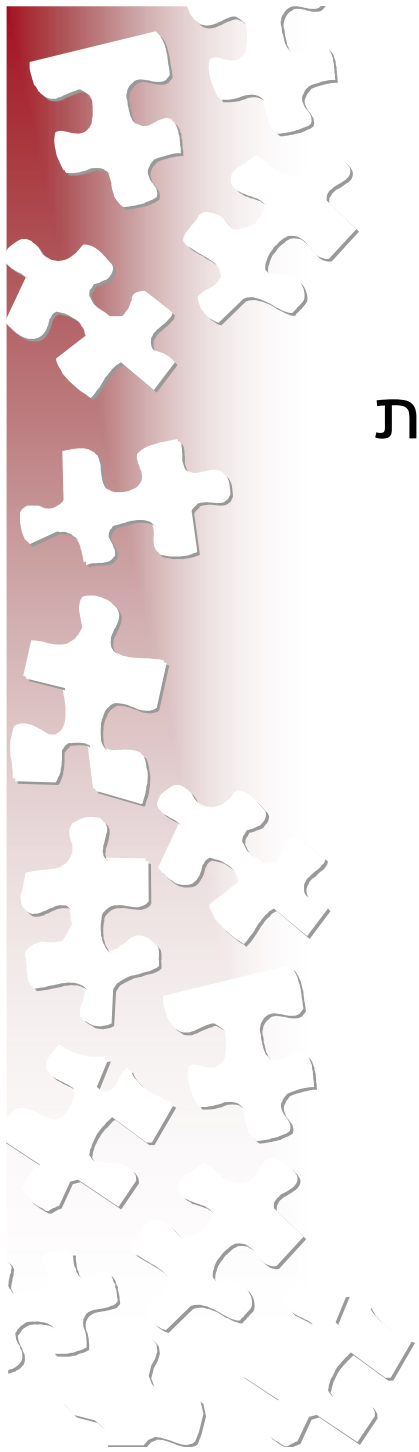
המחלקה להוראת המדעים  
מכון ויצמן למדע

**קווים מנחים להוראת תכנות**  
**מונחה עצמים למתחילים**  
**ד"ר נוע רגוניס**

**כנס המורים הארצי – המכללה האקדמית בית ברל**  
חנוכה תשס"ז – 17/12/06

# המעבר לתכנות מונחה עצמים

- ◀ בשני העשורים האחרונים מתרחש מעבר לתכנות מונחה עצמים.
- ◀ המעבר נוצר בעיקר כצורך לתת מסגרת הולמת לפיתוח מערכות תוכנה גדולות בתעשייה.
- ◀ השינוי בשטח התעשייה גורר עימו את השאלה האם השינוי צריך לחול גם באקדמיה ויותר מכך גם בשנים שקודמות לה – בבית הספר התיכון.



# תכנות מונחה עצמים באקדמיה ובתיכון

◀ באקדמיה בעולם יש הסכמה די נרחבת על כך שיש ללמד תכנות מונחה עצמים.

◀ השאלה המרכזית היא מתי? (שאלת הפרדיגמה הראשונה)

האם מן ההתחלה (גישת עצמים תחילה) או האם בהמשך.

◀ ההתלבטות היא דומה גם ביחס לתלמידי תיכון, אלא שכאן היא מעצימה מאחר ומדובר באוכלוסיה צעירה יותר וכזאת שלא בהכרח תתמחה במקצוע.

# על מה אדבר ?

## ◀ אם בוחרים בגישת עצמים תחילה אזי:

◀ מה הם הדגשים המרכזיים בהוראה

◀ מה הוא הסדר המומלץ להוראת התכנים

◀ אילו כלים דידקטיים כדאי לאמץ

◀ באילו דוגמאות כדאי לבחור בשלבים שונים של ההוראה

## ◀ מה המשמעות של דחיית ההוראה של עצמים

או: עד להיכן ניתן לדחות את הוראת העצמים כך שעדיין גישת תכנות מונחה עצמים תהיה המנחה בתהליך פתרון בעיות.



# על מה דברי מתבססים ?

## ◀ מחקר הדוקטורט שלי במכון ויצמן למדע בהנחיית פרופ' מרדכי בן-ארי

◀ לימדנו את התכנים של יסודות 1 בגישת עצמים תחילה לתלמידי כיתה יוד (שנתיים).

◀ ניהלנו מחקר עומק מלווה שבחן את הבנת המושגים על ידי התלמידים.

## ◀ סקר מעמיק של הנעשה בעולם בתחום

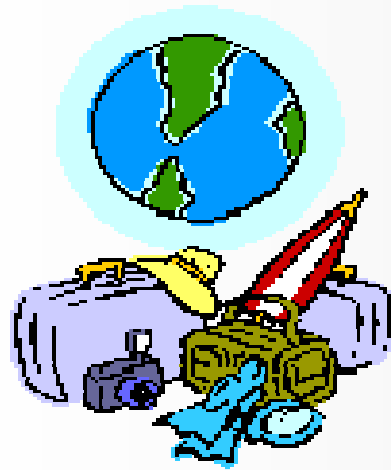
◀ הוראת התכנים לפרחי הוראה במכללת בית-ברל

## ◀ הוראת מורים בהשתלמויות

◀ כ- 250 מורים במהלך השנתיים האחרונות

# מה הוא תכנות מונחה עצמים

הבסיס התפיסתי של תכנות מונחה עצמים הוא:  
ככה העולם מאורגן

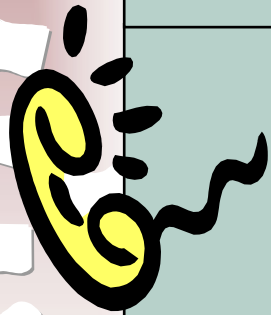


# דוגמאות לטיפוסים



טיפוס	קו טלפון
תכונות	מספר קו שם הבעלים תפוס
פעולות	שנה מצב תפוס האם תפוס?

טיפוס	שיחת טלפון
תכונות	קו מתקשר - מטיפוס קו טלפון שם מתקשר קו מקבל - מטיפוס קו טלפון שם מקבל שעת התחלה שעת סיום
פעולות	סיים שיחה החזר משך זמן שיחה החזר מחיר שיחה (לפי מחיר לדקה)



# דוגמאות לעצמים

יצירת שיחת הטלפון  
תעדכן את תכונות הקווים  
המעורבים להיות תפוסים



שיחת טלפון בין אלמו גוניור לעוגילידה	עצם
קו מתקשר: קו הטלפון של אלמו שם מתקשר: אלמו גוניור קו מקבל: קו הטלפון של עוגי שם מקבל: עוגילידה שעת התחלה: 8:15 שעת סיום:	תכונות

קו הטלפון של אלמו מטיפוס קו טלפון	עצם
מספר קו: 123456 שם הבעלים: אלמו תפוס: כן	תכונות

קו הטלפון של עוגי מטיפוס קו טלפון	עצם
מספר קו: 888888 שם הבעלים: עוגי תפוס: כן	תכונות



# סיווג טיפוסים: פשוט / מורכב

**טיפוס מורכב**  
התכונות שלו מטיפוסים (אחרים) שפותחו

טיפוס	שיחת טלפון
תכונות	<b>קו מתקשר</b> - מטיפוס קו טלפון שם מתקשר <b>קו מקבל</b> - מטיפוס קו טלפון שם מקבל שעת התחלה שעת סיום
פעולות	סיים שיחה החזר משך זמן שיחה החזר מחיר שיחה (לפי מחיר לדקה)

**טיפוס פשוט**  
התכונות שלו מטיפוסים בסיסיים

טיפוס	קו טלפון
תכונות	מספר קו שם הבעלים תפוס
פעולות	שנה מצב תפוס האם תפוס?

# סיכום מושגים מרכזיים בתמ"ע

◀ **טיפוס** - תבנית תאור כללית לבניית עצמים (תכונות ופעולות מאפיינות)

◀ **עצם** - מקרה פרטי, מופע של טיפוס (תכונות וערכים, ניתן להפעיל את הפעולות המוגדרות בטיפוס שלו)

◀ **תכונה** - מוגדרת בטיפוס באופן כללי, מקבלת ערך עבור עצם (ערך תכונה)

◀ **פעולה** - מוגדרת בטיפוס באופן כללי, מופעלת על עצם מן הטיפוס

◀ **הפעלת פעולה** - על עצם לא על טיפוס

◀ **בניית עצם** - טיפוס הוא תבנית, על פי התבנית בונים עצם, עצם הוא ישות

◀ **מצב עצם** - ערכי התכונות של עצם ברגע נתון

◀ **טיפוס יכול להיות מוגדר בעזרת טיפוס אחר** (שימוש/הכלה, הורשה)

# פתרון בעיות בתמ"ע

◀ כאשר אנו ניגשים לפתרון בעיה בתמ"ע אנו מזהים:

את **הישויות** הפועלות,

את **התכונות** של כל אחת מהן,

את **הפעולות** שניתן לבצע עליהן,

ואת **הקשרים** בין הישויות.

כל ישות היא טיפוס

טיפוס ממומש על ידי מחלקה בשפת התכנות

# פרויקט בתמ"ע

◀ בתמ"ע אנחנו מדברים על פרויקט הכולל מספר טיפוסים.

◀ כל טיפוס מייצג ישות מבין כלל הישויות שזוהו כמשתתפות בפתרון הבעיה.

◀ אבל טיפוס הוא תבנית, אז....

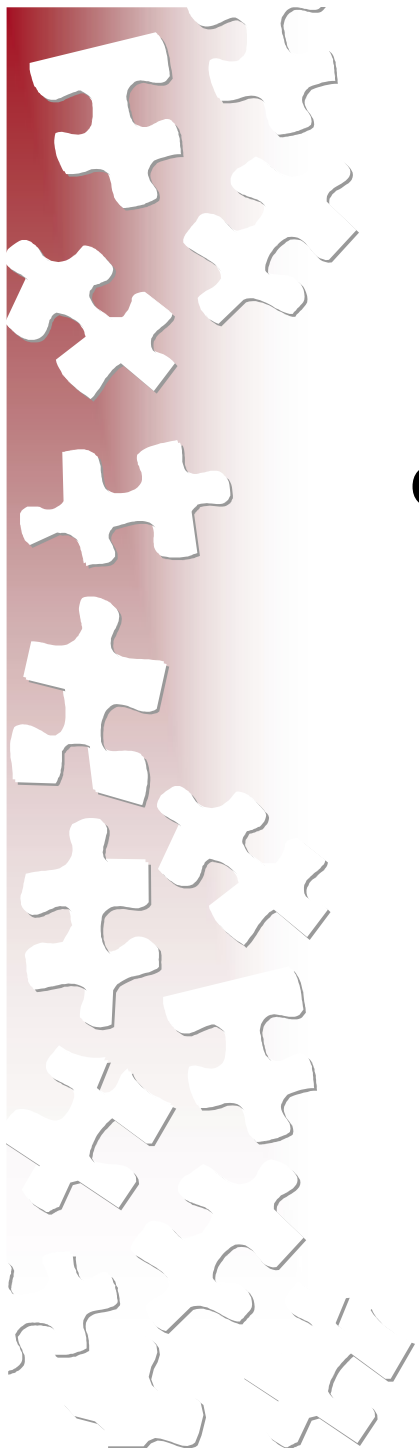
אם כך... היכן מתבצע האלגוריתם לפתרון הבעיה הנתונה?

# טיפוס ראשי – מחלקה ראשית

◀ האלגוריתם לפתרון בעיה נתונה יתבצע בטיפוס "אחר"

בטיפוס הראשי,

טיפוס **שאינו** מייצג ישות.



# טיפול ראשי – מחלקה ראשית

◀ הטיפול הראשי לא מייצג ישות – **מטרתו לממש את האלגוריתם הראשי** – הוא המנצח על תהליך הפתרון.

◀ לפיכך **לא נגדיר בו תכונות** ולא נגדיר בו פעולות המתאימות לטיפול המייצג ישות,

**אלא נגדיר בו פעולה ראשית בלבד.**

◀ מאחר וטיפול זה לא מייצג ישות גם **אין מקום** להגדיר על פיו עצמים.



```
class MainPhones
```

```
{
```

```
    // ללא תכונות
```

```
    public static void main(String[] args)
```

```
    {
```

```
        // יצירת עצמים מטיפוסים אחרים
```

```
        // הפעלת פעולות על עצמים
```

```
    }
```

```
}
```



# הפעולה הראשית בטיפול הראשי

◀ בפעולה הראשית יוגדרו העצמים המשתתפים בפתרון הבעיה ויופעלו עליהם פעולות.

◀ בטיפול המייצג ישות יש אוסף של פעולות, הפעלת הפעולות מתוך הטיפול הראשי תהיה בהתאם לצורך של פתרון הבעיה הנתונה.

◀ צרכי פתרון הבעיה יבואו לידי ביטוי בפעולה הראשית, היא תקבע את סדר ביצוע הפעולות בהתאם לנדרש.

◀ (הערה: כמובן שבטיפול זה יכולות להופיע עוד פעולות – כלומר, חלוקת בעיה לתת-בעיות, אלא שהן סטטיות – לא מופעלות על עצמים).



# מה שאפשר לעשות אך לא מומלץ לעשות !!!

◀ **מבחינת השפה** אין מניעה שלכל טיפוס תהיה פעולה ראשית, אך לדעתי מוטב לבנות טיפוס מיוחד לצורך הפעלת התכנית – ורק בו תוגדר הפעולה הראשית.

◀ **מבחינת השפה** ניתן גם לייצר עצמים מן הטיפוס הראשי, שהוא טיפוס לכל דבר, וניתן גם להגדיר בו פעולה בונה – אך לדעתי אין לכך הקשר הגיוני במסגרת תמ"ע.



# ההגדרות שלי לסוגים של טיפוסים

## טיפוס המייצג ישות

טיפוס פשוט (טיפוס שהתכונות שלו בסיסיות)

טיפוס מורכב (טיפוס שיש לו תכונות מטיפוסים אחרים  
שהגדיר המשתמש)

## טיפוס ראשי

אינו מייצג ישות

בו מתבצע האלגוריתם הראשי

בו מוגדרים העצמים ומופעלות עליהם פעולות



# ההגדרות שלי לסוגים של מחלקות

◀ **מחלקה ממשלת טיפוס** (המייצג ישות)

◀ **טיפוס פשוט** (טיפוס שהתכונות שלו בסיסיות)

◀ **טיפוס מורכב** (טיפוס שיש לו תכונות מטיפוסים אחרים שהגדיר המשתמש)

◀ **טיפוס ראשי / מחלקה ראשית / תכנית**

◀ **פרויקט מכיל אוסף של טיפוסים לפתרון בעיה**  
**בתוכם טיפוס ראשי אחד**

◀ **מחלקת שירות** היא מחלקה של פעולות סטטיות העוסקות באותו "עניין". היא לא מייצגת ישות וגם אינה מבצעת אלגוריתם ראשי לפתרון בעיה.

# הוראה בגישת עצמים תחילה מה זה אומר?

◀ ללמד במקביל את כל מכלול המושגים שהודגשו. המושגים קשורים זה בזה ולא ניתן להתייחס לאף אחד מהם בלי להתייחס אל האחרים.

◀ להתחיל עם מושגי הבסיס של תמ"ע ולדחות את מושגי הבסיס באלגוריתמיקה. (עדיין ניתן להקיף את התכנים והרעיונות המרכזיים המוגדרים בתכנית הלימודים "יסודות מדעי המחשב 1-2").

◀ עקב מורכבות המושגים והעיסוק במושגים מופשטים, מהותי מאד לבחור בכלים פדגוגיים מתאימים. במקרה שלנו כאלה המתאימים לתלמידי תיכון.

# המטרות המרכזיות בהוראה

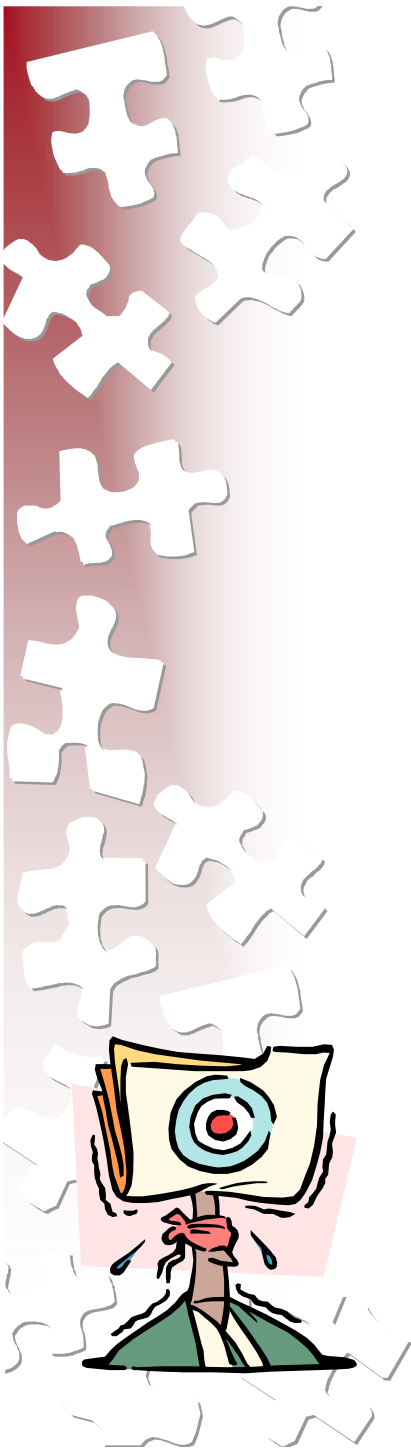
◀ **מושגי היסוד בתמ"ע שאת הבנתם אפשר לדעתי להשיג ב "יסודות"**

◀ **מודולאריות** (חלוקת בעיה לטיפוסים שונים)

◀ **הכמסה** (עצם הוא כמוסה שיכול לשמש ערך של תכונה בעצם אחר)

◀ **הסתרת מידע** (עבודה עם ממשקי טיפוסים, הקפדה על פעולות קובעות ומאחזרות)

◀ **זרימה של תכנית** (חשוב להתייחס אליה דווקא בגלל שהדגש הוא על טיפוסים המייצגים ישויות)



# המטרות המרכזיות בהוראה

## ◀ המשמעות היישומית של הקניית מושגים אלו

- ◀ בהינתן בעיה התלמידים יזהו את הטיפוסים הנדרשים להגדרה לצורך פתרונה.
- ◀ התלמידים ידעו להגדיר טיפוסים פשוטים ומורכבים בשפת התכנות.
- ◀ התלמידים ידעו להשתמש בטיפוס נתון שלא פיתחו בעצמם על פי הממשק שלו.
- ◀ התלמידים ידעו להגדיר תכנית ראשית המבצעת את האלגוריתם הראשי לפתרון הבעיה תוך שימוש בטיפוסים שהגדירו.



# שלבים עיקריים בהוראת תמ"ע

## 1. מתחילים לדבר על עצמים

◀ תכונות, ערכי תכונות, מצב העצם, אחזור ערכי תכונות (פעולות מאחזרות), שינוי ערכי תכונות (פעולות קובעות), שינוי מצב העצם.

## 2. לעצמים רבים יש מכנה משותף – תכונות ופעולות ← טיפוס

◀ טיפוס הוא תבנית, על פי טיפוס בונים עצמים, לעצמים יש את התכונות המוגדרות בטיפוס שלהם, על עצמים ניתן להפעיל את הפעולות המוגדרות בטיפוס שלהם.



# שלבים עיקריים בהוראת תמ"ע (המשך)

## 3. לעצם יש מזהה

- ◀ אל עצמים מתייחסים בעזרת המזהה שלהם, לכל עצם יש מזהה ייחודי משלו.
- ◀ זהירות: לא לדבר על שם העצם, ולא לבלבל עם תכונה שהיא "שם".

## 4. עצם ↔ טיפוס

- ◀ אין עצם ללא טיפוס, אם יש רק טיפוסים אי אפשר לעשות כלום, טיפוסים הם תבניות, רק אחרי בניית עצמים ניתן להפעיל עליהם פעולות.





# שלבבים עיקריים בהוראת תמ"ע (המשך)

## 5. בניית עצמים

- ▶ התהליך המרכזי ביותר.
- ▶ עצם נבנה על ידי הפעלת פעולה בונה.
- ▶ בתחילת ההוראה יש לדבר על השלבים הבאים בתהליך יצירת העצם: נבנה עצם שיש לו תכונות על פי הגדרת הטיפוס שלו, התכונות מקבלות ערכים לפי פרמטרים של הפעולה הבונה.



# שלבים עיקריים בהוראת תמ"ע (המשך)

## 6. מצב העצם

- ◀ אוסף התכונות וערכיהן.
- ◀ מצב העצם משתנה על ידי ביצוע פעולות.
- ◀ אין משמעות לפעולות שאינן מתייחסות לתכונות העצם – כי הן יכולות להיות בכל מקום.



# שלבים עיקריים בהוראת תמ"ע (המשך)

## 7. שילוב טיפוס מורכב

- ◀ חשוב להציג טיפוס מורכב כבר בשלב ההכרות עם תמ"ע.
- ◀ ללא טיפוסים מורכבים אין משמעות לתמ"ע ולכן חשוב לתת את התחושה הנכונה של המסגרת כבר בהתחלה.
- ◀ כאשר יש התקדמות לשלב התכנות הטיפוס המורכב יוצג מעט מאוחר יותר. לאחר הגדרת המחלקה הראשית.



# שלבים עיקריים בהוראת תמ"ע (המשך)

## 8. שילוב טיפוס ראשי / מחלקה ראשית / תכנית

- ◀ בשלבי המימוש יש להקדים הגדרת מחלקה ראשית להגדרת מחלקה של טיפוס מורכב.
- ◀ שלב זה מהותי לתמיכה בהבנת תהליך בניית העצמים, הבנת מזהה העצם, הבנת תהליך הפעלת הפעולות על העצמים, הבחנה בתהליכים המביאים לשינוי במצב העצם.



# כלים פדגוגיים בהוראת מושגי היסוד של תמ"ע

1. הצגת עצמים בעזרת דיאגרמות.
2. הצגת טיפוסים בעזרת דיאגרמות.
3. תרגול עצמי "ניר ועפרון" בהגדרת עצמים וטיפוסים.
4. הצגת כל התהליכים בעזרת סביבת ויזואליזציה כדוגמת BlueJ תוך מתן דגש על כל מושגי היסוד.
5. תרגול עצמי בסביבת הויזואליזציה (ללא תכנית ראשית).

# חשיפה למבנה מחלקה – התחלת רכישת מיומנויות תכנות

1. שימוש בסביבת הויזואליזציה מקנה היכרות ישירה ואינטואיטיבית עם מבנים בשפה (מבנה כותרת של פעולה, פרמטרים ועוד...).

2. הכרות מצפייה בטיפוס מלא שהוגדר מראש וניתן לתלמידים (זה שתרגלו איתו בניית עצמים...).

3. בגוף הפעולות יהיו רק פעולות אלמנטריות של השמה או ביצוע פעולות חישוב אלמנטריות.





# התפתחות מיומנויות תכנות בסיסי

▶ **פיתוח שלב 1:** שינוי קל בפעולות קיימות (במקום +1 לבצע -1, ~~~ משהו כזה).

▶ **פיתוח שלב 2:** מתן כותרת של פעולה ובקשה להשלים את גוף הפעולה.

▶ **פיתוח שלב 3:** הגדרה מלאה של פעולה.

▶ **פיתוח שלב 4:** הגדרה של פעולות קובעות ומאחזרות.

▶ **פיתוח שלב 5:** הגדרה של פעולות בונות חלופיות.

▶ **פיתוח שלב 6:** הגדרה של פעולה בונה המשלבת קלט מן המשתמש.

▶ **פיתוח שלב 7:** הגדרת טיפוס פשוט מלא.

# התפתחות כלי תכנות בסיסים

◀ **שלב 1:** משפטי השמה.

◀ **שלב 2:** ביטויים חשבוניים.

◀ **שלב 3:** משפטי פלט.

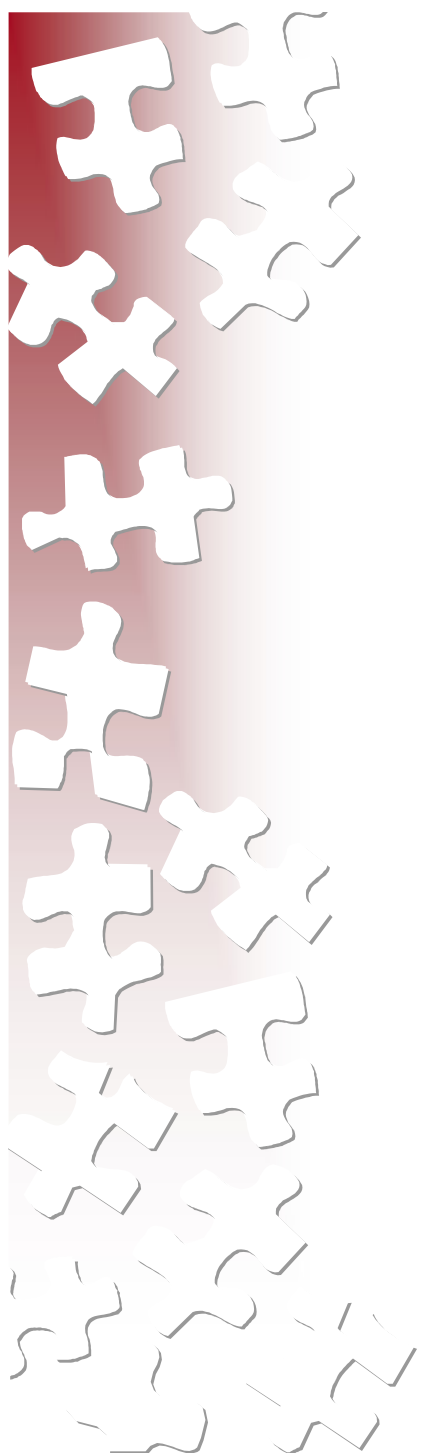
◀ **שלב 4:** משפטי קלט.

◀ **שלב 5:** הוראות תנאי.

◀ **שלב 6:** לולאות (← דווקא אתגר... לתת הקשר מתאים 😊).

◀ **שלב 7:** מערכים של טיפוסים בסיסיים.

◀ **שלב 8:** מערכים של עצמים.





# ואחר כך יש זמן ל...

1. מערכים דו-מימדים

2. פעולות סטטיות

3. מחלקת שירות (מחלקה שהיא אוסף של פעולות סטטיות)

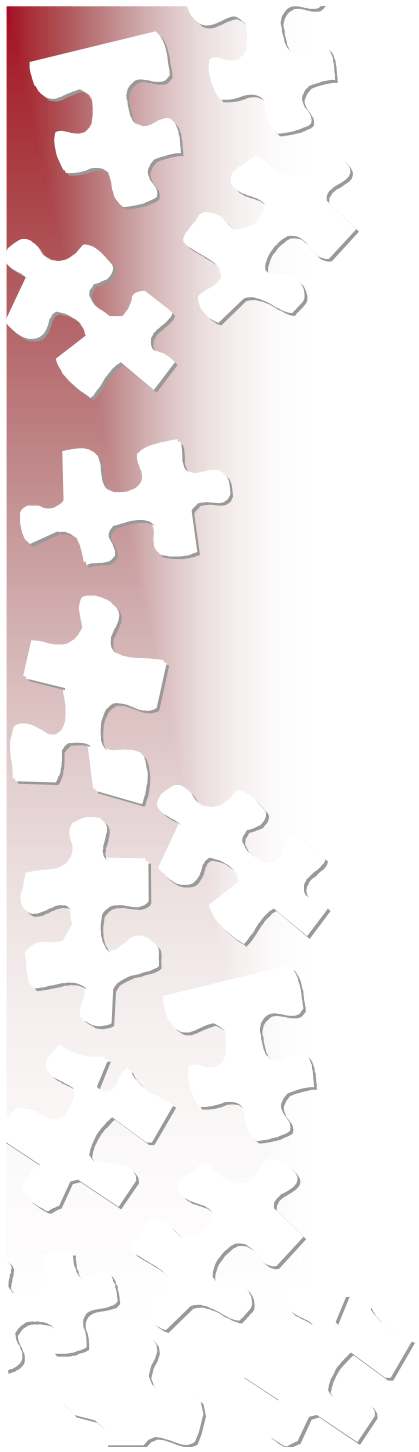
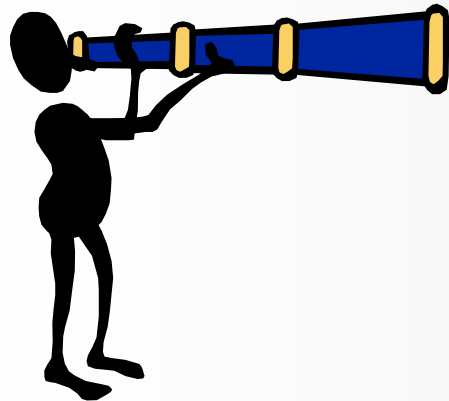
וגם המחרוזות אי שם באמצע...



# שימו לב: כלי התכנות הנרכשים "בלי להרגיש"

1. פרמטרים

2. פעולות



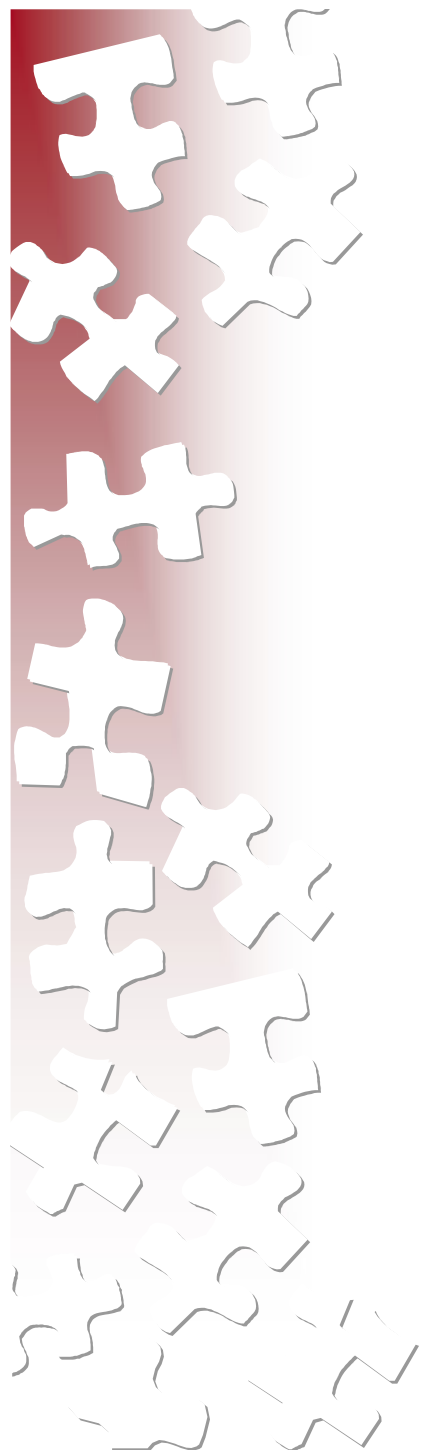


# התפתחות שימוש בסוגי מחלקות

- ◀ **שלב 1:** מחלקה המממשת טיפוס פשוט המייצג ישות.
- ◀ **שלב 2:** מחלקה ראשית ובה בניית עצמים מטיפוס פשוט.
- ◀ **שלב 3:** מחלקה ראשית ובה עצמים משני טיפוסים פשוטים.
- ◀ **שלב 4:** מחלקה המממשת טיפוס מורכב.
- ◀ **שלב 5:** מערך של עצמים במחלקה ראשית.
- ◀ **שלב 6:** מערך של עצמים כתכונה בטיפוס מורכב.

**מהותי! חוץ מאשר בשלבים הראשונים  
כל הזמן לעבוד עם תכנית ראשית**

# ואיך כל זה מסתדר עם תכנית הלימודים הקיימת ???





## ציטוטים מתוך תכנית הלימודים (1)

◀ הקורסים "יסודות מדעי המחשב 1 ו-2" הם קורסי חובה ראשונים ללומדים מדעי המחשב בחטיבה העליונה.

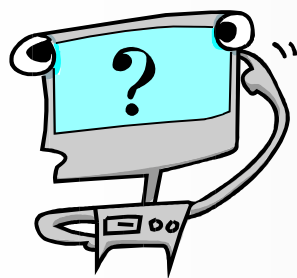
◀ מטרת התכנית היא ללמד מושגי יסוד של מדעי המחשב וכן את עקרונות המבנה של מערכות ממוחשבות.

◀ התכנית מדגישה עקרונות שעומדים במבחן הזמן לעומת מושגים תלויי טכנולוגיה שצפויים להתיישן במהירות.

## ציטוטים מתוך תכנית הלימודים (2)

▶ מטרת התכנית אינה הכשרה מקצועית ואף לא הקניית מיומנויות טכנולוגיות, כגון לימוד שפת תכנות על בוריה.

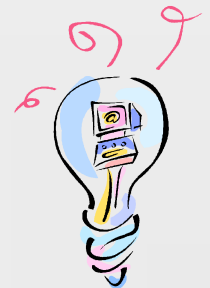
▶ התכנית מקנה ללומדים יכולת חשיבה, ומיומנויות לניתוח ופתרון של בעיות.



# עקרונות היסוד העומדים בבסיס החשיבה שלי על המעבר

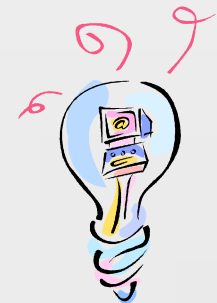
1. הוראת תמ"ע מן ההתחלה מאפשרת לעמוד בכל עקרונות היסוד של תכנית הלימודים הקיימת.

2. אי אפשר להתעלם מן הפרדיגמה החדשה ומן השפות החדשות בפן הענייני וגם בפן של דעת הקהל (גיוס תלמידים והורים).



# עקרונות היסוד העומדים בבסיס החשיבה שלי על המעבר (המשך)

3. ניתן לבחור בגישת "עצמים תחילה" כפי שהצגתי כאן, או בגישת "עצמים קצת אחרי ההתחלה" (אחרי הקניית בסיס של תכנות **ולפני** התקדמות לפתרון בעיות מורכבות).



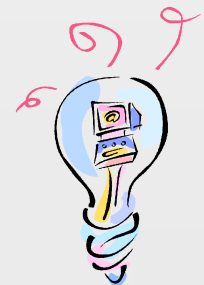


# עקרונות היסוד העומדים בבסיס החשיבה שלי על המעבר (המשך)

4. הדגש על פתרון בעיות אלגוריתמיות  
"עובר מקום" ויידון באשר להגדרת פעולות  
ולא באשר לניתוח כולל של בעיה.

כלומר השלבים המקובלים בפיתוח פתרון לבעיה  
ישמרו עבור פיתוח פעולה:

ניתוח קלט/פלט, דוגמאות מייצגות, פיתוח אלגוריתם  
מילולי, פיתוח קוד בשפת מחשב.



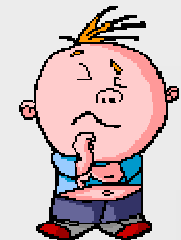
# ועוד הגיגים על המעבר

שפות התכנות המתקדמות מאפשרות מגוון של כלי תכנות, מבני בקרה, וספריות בנויות.

אין מקום לכלול בקורסי יסוד אלו הפונים לכלל האוכלוסייה את השפע הקיים בשפות.

יש להסתפק בכלים ההכרחיים והתואמים את תכנית הלימודים הקיימת.

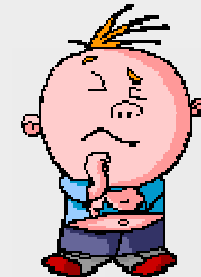
מעט הרחבות ישולבו במקום בו יש לכלים נוספים בשפה יתרון.

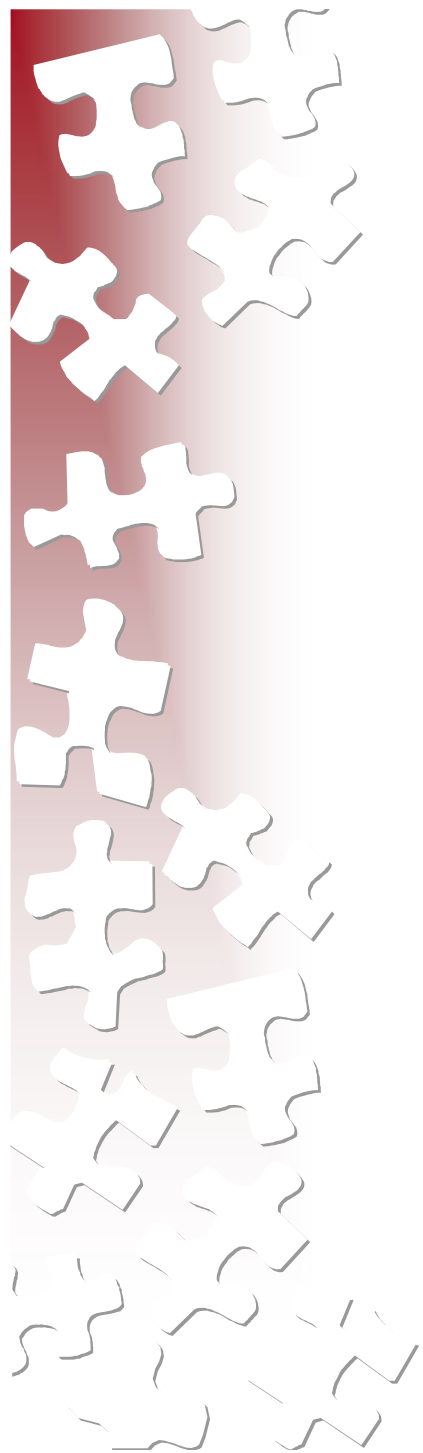


# ועוד הגיגים על המעבר (המשך)

◀ לדעתי אין צורך לדון בנושא בקרת גישה לפעולות או משתנים.

ניתן להסתפק בבקרת גישה private עבור תכונות כדי להדגיש את עקרון הסתר המידע. ובכל היתר לעבוד עם ברירת המחדל של public.





# תודה וחג אורים שמח

