

# מבוא לתכנות מונחה עצמים

---

## שיעור 1 – היכרות ומבוא כללי

# על הקורס

- מטרה: הקניית יכולות תכנון ופיתוח מערכות תוכנה מורכבות
- אמצעים: כלים טובים ועבודה קשה.

- מרצה אחראי: פרופ' בעז בן משה [benmo@g.ariel.ac.il](mailto:benmo@g.ariel.ac.il)

- מרצים נוספים: ד"ר אראל סגל-הלוי, אודי לביא

- תירגולים: רויטל מרבל, ספי ארליך

- אתר הקורס:

<https://moodle.ariel.ac.il/course/view.php?id=57796>

- שעות קבלה: בועז - יום רביעי 9-11, בחדר 5.2.11  
אראל – אחרי השיעור, בכיתה או בחדר 11.3.13D.

# שיעור 1 – סדר יום

- חלק א. היכרות
- חלק ב. מבוא לקורס: מתיכנות להנדסת-תוכנה
- חלק ג. תיכנון מונחה עצמים: תרגיל כיתה
- חלק ד. מטלות הקורס, מטלה 0

# שלום שנה ב

■ מי מתכנת ב: Java? C? C++? C#? Basic? Perl?  
PHP? Python? HTML? Javascript? Shakespeare

■ מי עובד ב: Linux? Mac? Windows

■ מי אוהב לתכנת? מי מתכנת בעבודה? בשעות הפנאי?

■ מה אתם זוכרים ממבוא לחישוב:

■ מחלקות, אובייקטים, קובצי טקסט, טיפול בשגיאות?

■ מה אתם זוכרים ממבנה נתונים:

■ עצים בינאריים, Hashtable, Iterators

תכנות מונחה עצמים - שיעור 1

# מתיכנות להנדסת תוכנה

- מערכות תוכנה גדולות הן **מורכבות** מאד מאד  
מיליוני שורות קוד
  - אנשים מצפים מתוכנה להיות חדשנית **ועדכנית**  
אי ודאות גדולה (לעומת הנדסת בניין למשל)
  - אנשים מצפים מתוכנה להתמודד עם **תקלות**  
טעויות משתמש, תקלות חומרה, תקלות תיכנות
- תיכנון נכון משפיע באופן דרמטי על **זמן** הפיתוח **ואיכותו**.  
בפיתוח תוכנה **זמן = כסף**.

# התמודדות עם מורכבות של תוכנה

- 1) פירוק לרכיבים קטנים.
- 2) הפרדה בין רכיבים שונים – כל רכיב עומד בפני עצמו.
- 3) תיעוד כל רכיב בנפרד וכולם יחד.
- 4) בדיקת כל רכיב בנפרד וכולם יחד.
- 5) תיכנות "מתגונן" -- להתכונן מראש לשגיאות של מפתחים אחרים.  
"לפני עוור לא תתן מכשול".
- 6) שימוש ברכיבים קיימים ובדוקים – קוד פתוח. לימוד תמידי.  
"ישמע חכם ויוסף לקח".

# עקרונות תיכנון מונחה עצמים

- (1) עצם = דבר שמשלב מצב עם התנהגות.  
מחלקה = "מפרט" לייצור עצמים עם  
מצבים נפרדים ו התנהגות זהה.  
לכל מחלקה מצמידים מייד תיעוד ובדיקות.
- (2) הסתרת מידע –  
הסתרת המורכבות של העצם.  
מניעת שינויים לא צפויים.
- (3) ירושה – יצירת עצמים ספציפיים מתוך עצמים כלליים.  
אפשר לכתוב אלגוריתם על מחלקת-בסיס ולהשתמש בו  
גם על מחלקות יורשות.

# נושאים עיקריים שנלמד בקורס

- (1) תיכנון – הפיכת בעיה לאוסף של מחלקות.
- (2) כתיבת קוד קל לתחזוקה: תיעוד ובדיקות.
- (3) שימוש בקוד קיים – קוד של מתכנתים אחרים / קוד פתוח.
- (4) שאלות בפורומים מקצועיים.
- (5) עבודת צוות, תיכנות בזוגות.

## נושאים משניים:

- (1) תכונות מתקדמות של שפת Java
- (2) אפליקציות Web, שפת Javascript, HTML



# תרגיל כיתה

אתם חלק מצוות העוסק בבניית לומדה לחקר פונקציות.  
תפקידכם בצוות: לבנות מחלקה שמתארת **פולינום**.

מה פולינום יודע לעשות?

● חיבור.

● כפל.

● ערך בנקודה.

● נגזרת.

איזה משתני-מצב יש לפולינום?

● מקדמים וחזקות.

● א': אוסף **מונומים**:

● כל מונום = מקדם וחזקה.

# שלבי עבודה

1. תיכנון: תרשים מחלקות.  
לכל מחלקה, כותבים:
2. מחלקת-בדיקה.
3. המחלקה עצמה.
4. בנאי.
5. שדות פרטיים.
6. מתודת toString.
7. מתודות נוספות: (א) מתודות קוראות, (ב) מתודות כותבות.
8. קריאת קוד – מציאת באגים של המתכנת.
9. הגנת קוד - התגוננות משגיאות של המשתמש.
10. סידור הקוד – refactoring. כל הבדיקות נשארות!

# מטלות הקורס - כללי המשחק

- תקבלו מטלה מתגלגלת – מטלה אחת בחמישה שלבים.
- בכל שלב תשפרו את המטלה הקודמת.
- כל מטלה 10%. בחינה סופית 50%.
- המטלות ייבדקו ע"י המתרגלים במהלך התירגול.
- אסור להעתיק (מסמכים), מותר לדבר (קבוצות דיון).
- חובה לציין **מקור** של כל קוד שמתמשים בו!
- מותר לשאול שאלות בפורומים. אסור לקנות תרגילים.
- על כל יום איחור "משלמים" 3 נקודות.
- חריגים רגילים: מילואים, חופשת לידה, מחלה.

# מטלה 0

- חוזרים לכושר: מתכנים מטלה בסיסית עובדת –  
יהיה הרבה מקום לשיפורים בעתיד.
- צריך להבין את המטלה, להתנסות, לבצע חיפוש של פתרונות קיימים.  
מטלה בזוגות.
- מסמך כללי כתיבת קוד – במודל. יוסבר בתירגול.
- הגשה: בעוד שבועיים, בדיקה בתרגול!

# מה למדנו היום

- "שלום שלום"
- מסגרת הקורס – כללי המשחק
- מוטיבציה לקורס
- חזרה: מחלקות, אובייקטים, קבצים, טיפול בחריגים.
- מטלה 0

# עד השיעור הבא

- סביבת פיתוח נוחה ועובדת (java & eclipse),
- מומלץ להתקין Linux (לא חובה)
- מטלה 0: תתחילו לתכנן ולממש.
- אם יש שאלות – תשאלו במודל, בשיעור הבא נערוך דיון בנושאים שיעלו.