

תכנות מתקדם 1

מבוסס על תרגולים של מר רוכלין איגור, אוניברסיטת בר-אילן 2013.

שיעור 1

הקדמה על שפת *Java*, הסבר על המתרגם, ה-*Interpreter* ועל תהליך הקומפילציה של תוכניות *Java* בעבר.

שיעור 2

כיום, *Java* עובדת טיפה שונה בעזרת הרצת ה-*Bytecode* של *Java*, בין השאר בעזרת מנגנון *JIT*:
JIT: Just in time.

תחביר *Java* דומה מבחינת תחביר (Syntax) לשפת *C++*. ב-*Java* עובדים עם אובייקטים: כל דבר הוא אובייקט.

ההבדל בין *Class* ל-*Instance*: *Class* הוא התבנית של האובייקט, בעוד ש-*Instance* הוא הופעה חדשה של האובייקט עצמו עם ערכים בתוכו.

ב-*Java*, המבנה ההיררכי של ספריות נקרא *Package*. על מנת להשתמש בספרייה אחרת (לצורך הפעלת מתודות למשל), נכתוב `import packageName.ClassName` בתחילת הקוד. ספריית ה-*src* דומה בתפקידה ל-*namespace* בשפת *C++*.

איך מגדירים עבור קוד להיות שייך לספרייה מסויימת? בתחילת הקוד, נכתוב `package packageName`. נהוג להתחיל שמות ספריות עם אות קטנה.

אפשר ליצור מחלקות (*Classes*) שונות בעלות אותו שם, אך תחת ספרייה (*package*) שונה.

Scope: קטע הקוד הנוכחי, שלתוכנית שרצה יש גישה אליו. דבר זה קובע בין השאר את אורך החיים של משתנים, או את המתודות המוכרות לדוגמה.

depracted: מציין אזהרה על מתודה מסויימת, אשר אומר כי ישנה דרך חדשה וטובה יותר לבצע את הפעולה. פעולה זו נשארה זמינה לצורך תאימות לאחור ואין להשתמש בה הלאה בפיתוח עתידי. מסומן בקו אמצעי.

Static: משתנים ומתודות המוגדרים להיות סטטיים, נוצרים בתחילת הריצה של הפרוייקט במקום מיוחד בהתחלת הזכרון.

This: כאשר נמצאים ב-*Scope* של מתודה *A*, אפשר להתייחס למשתנים השייכים למתודה זו ע"י `this.var`. זוהי צורה שקולה.