



תכנות מונחה עצמים

אובייקט - מקום בזיכרון שיש לו סדרק וקיים אליו קישור (רפרנט) טיפוס חפץ
שדות - המשתנים של האובייקט

בא' - הפונק' שיוצרת instance של האובייקט

מתודות - הפונק' של האובייקט

מצב פנימי (state) - מאופיין ע"י ערכי השדות

members - שדות ומתודות

Access Modifiers

לשפה	מתודה	מחלקה	
ניתן לגשת מכל מקום	ניתן לקרוא/לערוך מכל מקום	ניתן לרשת מכל מקום	public
ניתן לגשת בתיק package	ניתן לקרוא/לערוך בתיק package	-	protected
ניתן לגשת רק באובייקט	ניתן לעשות רק בתיק אובייקט	-	private

reference - מצביע לאובייקט, נשמר ב heap.

Garbage Collector - מוסעל מפי עצם בזמן הריצה באופן אוטומטי ומחק

unreferenced - זיכרון

String - טיפוס מורכב הקיים בשפה. מכיל מתודות רבות כגון length(), indexOf(char) ואף

מאריך - טיפוס מורכב נוסף הקיים בשפה. אובייקט length (שמר בתור שדה (public))

ומכיל שתי מתודות: equals (השוואה ע"פ רפרנט) ! toString

חמשק - Interface

מאפשר הפרדה בין יכולת לבין לממש.
ממשק מכיל חתימות של הסוג

כל class שמממשת ממשק, מתחייבת לממש את כל החתומות שלו.


דוגמה:

```
public interface Shape {
    public double area();
}
```

```
public class Circle implements Shape {
    ;
```

פולימורפיזם: תאפשרות עקרון עמחלקה Circle לבזמה בתור Shape מועד את הצורך עריבוי מתודות. קומפילר עושה זאת באמצעות *upcasting*

Static - מאפשר גיעה עמיתודה / שדה בלי יצירת instance שלו.
מתודות סטיות עמו יכולות לעשת עמיתודה / שדות שונים סטית.

מחלקה	מתודה	שדה	final 
עמו יכולה שיתשו איתה	עמו יכולה עהיפרם	עמו יכולה עשיות ערק	

Encapsulation

במתודות getter ! setter של שדות מאיפוסים מורכבים
יש עי"צ instance חש ורק אז עתחזירו/עשהימו

הורשה Inheritance

כאשר מחלקה יורשת מחלקה אחרת, היא מקבלת את כל תכונות האבא שלה. תורשה יוצרת היררכיה

^{implicit} **downcasting** - מצב לא חיוני. הפעלת מתודה של בן אל מופע של האב.
גם אם האב (האב) נטו מייצגים של הבן, תתקבל שגיאה קומפילציה.
ניתן לעשות downcast רגיל או ניתן להפסיד את מתודות הבן

אם אנשים downcast, לבן כלשהו של היפוס, התכנית תתקמפל. לבן אם לא ניתן לבצע את ה downcast תתקבל שגיאה זמן-ריצה

default - מופשר מיושם של מתודה ושירות בתיק ממשק. default void...
ניתן מאוחר אך רצוי לממש בתיק המיחלקות

הערה: ניתן להוסיף לממשק שדות סטטיים וסופרים (שניהם בודד), יריד באולו.

Abstract Classes

מחלקה שמכילה מתודות שאינן ממומשות. יל לבית abstract class...
לפני כל מתודה לא ממומשת לבית abstract

כל מחלקה שאינה אבסטרקטית חורשת מחלקה אבסטרקטית מתחייבת לממש את כל תחתיות שאינן מיושם באב

לא ניתן ליצור מופעים של abstract class. (קבל שגיאה קומפילציה אם ננסה לעשות בן.

super - קריאה לבטוי של האב. תיוב להעשיות כשורה היאושועה של הבטוי של הבן.

@Override - מצוין לפני פריסת מתודת האב במחלקת אחר מילדי.

Class Object

המחלקה הפי בסיסית כל מחלקה יורשת אותה. יש לה מתודות toString
! equals (המשווה ע"ש רפרנס)

instance of - בודדת בזמן ריצה האם אובייקט הוא מטיפוס מסוים
Song instance of Student
false

Static & Dynamic Binding

על זכר יש תמונת טיפוס קבוע.

ע"פ"ה יש שני טיפוסים:

- טיפוס סטטי: הטיפוס בהגדרת המשתנה. טיפוס קבוע.

- טיפוס דינמי: טיפוס העצם המוצג.

הטיפוס הפינאלי תיב להיות לגדרת של הטיפוס הסטטי.

מתי יש Static Binding

private - מתודת

static - מתודת

final - מתודת

public - ע"ש משתנים

Overloading

שתי מתודות בעלי שם זהה אך בעלות ארגומנטים שונים (מספיק סדר שונה)
הקומפילר ינסה לזהות אותן ומתודות שבהירות באופן זהיר קרוב
לטיפוס ה"אמיתי"

Delegation

הרעיון הוא שמחלקה תבצע את כל הפעולות בעצמה, היא מחזיקה
שדה שהוא class אחרת והיא מבצעת את הפעולות

Generic Programming

מעגן פיתוח תוכנה בו אלגוריתמים נכתבים באמצעות טיפוסים סוגגרו
בהמשך.