ת מחלקה
ת מחלקה

static variables

static methods

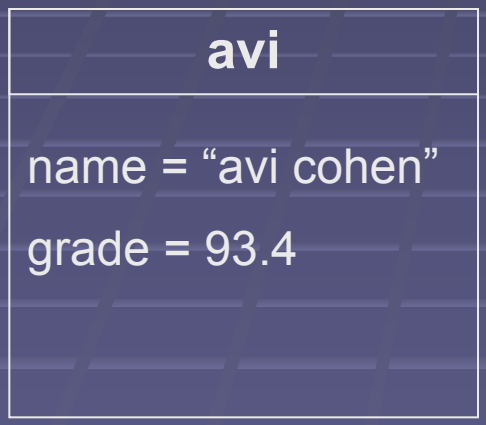
עד עתה – תכונות שייכות לעצם

```
public class Student {
    private String name;
    private double grade;

    public Student(String name, double grade) {
     this.name = name;
     this.grade = grade;
    }
}
```

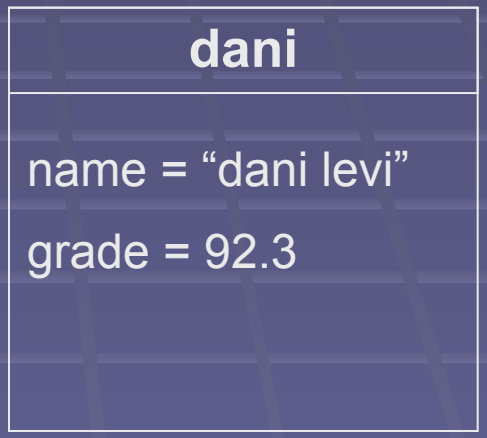
בית הספר

```
public class Student {
    private String name;
    private double grade;

    public Student(String name, double grade) {
     this.name = name;
     this.grade = grade;
}
```

Student avi = new Student(…);



Student dani = new Student(…);



ים

Student avi = new Student(…);

avi
name = “avi cohen”
grade = 93.4

Student dani = new Student(…);

dani
name = “dani levi”
grade = 92.3

התכונות שייכות לעצמים avi ו-dani

- **גישה פרטית מתוך העצם לערכים.**
- **אין קשר בין הערכים בשני העצמים.**
- **שניהם לפי תבנית המחלקה**

ר סידורי לתלמידים

- בית הספר מבקש לתת לכל תלמיד מספר סידורי באופן רצוף.
- ניצור לכל תלמיד תכונה חדשה 'מספר סידורי'.
- אבל !!! זו דרישה לקשר בין עצמים שונים מאותה מחלקה:
  - צריך לדעת מי היה התלמיד הקודם שנוצר.
  - צריך לדעת מה היה מספרו הסידורי.
  - צריך לספור באופן רצוף.

מחלקה

- מספר סידורי הוא אכן תכונה ייחודית לכל תלמיד.
- אבל **מונה המספרים הסידוריים** הינו משתנה המשרת את כל העצמים במחלקה.
  - הוא מאותחל ל-0 לפני שנוצר עצם כלשהו.
  - ערכו עולה ב-1 בזמן יצירת העצם.
  - ערכו מועתק למספר הסידורי של התלמיד.
- מונה המספרים אינו משוייך לתלמיד ספציפי.

מחלקה

משתנה המשרת את כל עצמי המחלקה
ואינו משוייך לעצם כלשהו, קרוי
איבר של מחלקה

מחלקה

משתנה המשרת את כל עצמי המחלקה
ואינו משוייך לעצם כלשהו, קרוי
איבר של מחלקה

הצהרה ב-Java:

```
public class Student {
   private static int counter = 0;
   private int id;

   private String name;
   private double grade;
```

```
public static int counte;
```

- ***private – מאפשר גישה מהמחלקה בלבד***
- ***static – מגדיר את המשתנה כמשתנה מחלקה***
- ***אתחול – כבר בשלב ההגדרה***

```
public static int counte;
```

- private – מאפשר גישה מהמחלקה בלבד
- static – מגדיר את המשתנה כמשתנה מחלקה
- אתחול – כבר בשלב ההגדרה

- Java יוצרת משתנה אחד עבור כל העצמים
- לכל עצם יש גישה אליו.
- כל עצם יכול לשנות את ערכו

ש במשתנה מחלקה

```
public Student(String name, double grade) {
 this.name = name;
 this.grade = grade;

 this.counter++;
 this.id = this.counter;
}
```

Student avi = new Student(…);

Counter = 1

avi
name = “avi cohen”
grade = 93.4
id = 1

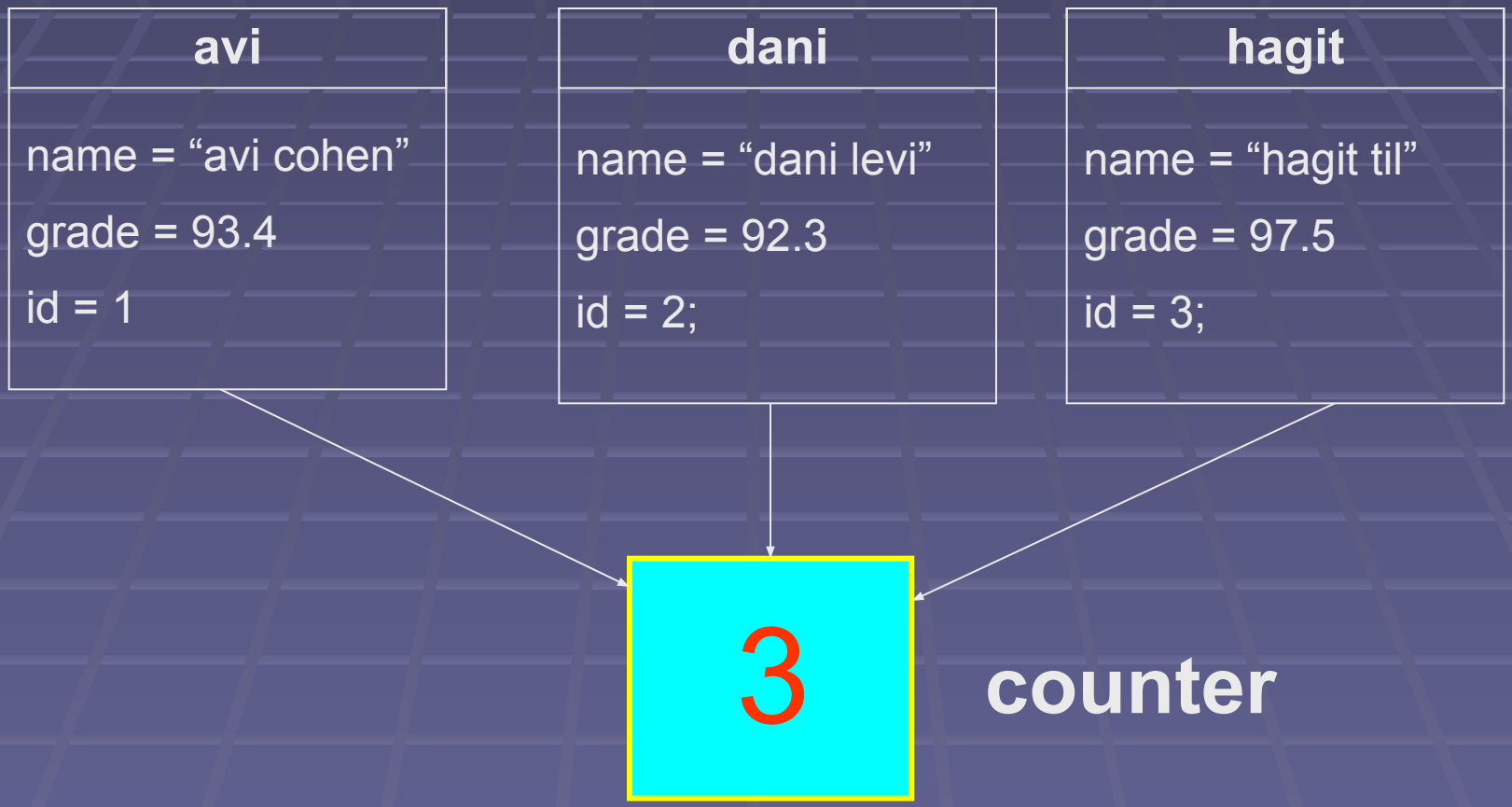
Counter = 2

Student dani = new Student(…);

dani
name = “dani levi”
grade = 92.3
id = 2;

Counter = 3

ה המדוייקת



יש עותק אחד בלבד של המשתנה counter

חלקה

- בג'אווה, הגישה לתכונות הינה באמצעות שיטות המשוייכות לעצם ספציפי.
- איך ניגש לתכונות מחלקה, שאינן משוייכות לעצם?
- באמצעות שיטות מחלקה.

()get

```
public static int getCounter() {
    return this.counter;
}
```

שיטות סטטיות נגישות לתכונות סטטיות בלבד !!!

()get

```
public static int getCounter() {
    return this.counter;
}
```

שיטות סטטיות נגישות לתכונות סטטיות בלבד !!!

אבל... איך משתמשים בשיטה ????

ש בשיטות מחלקה

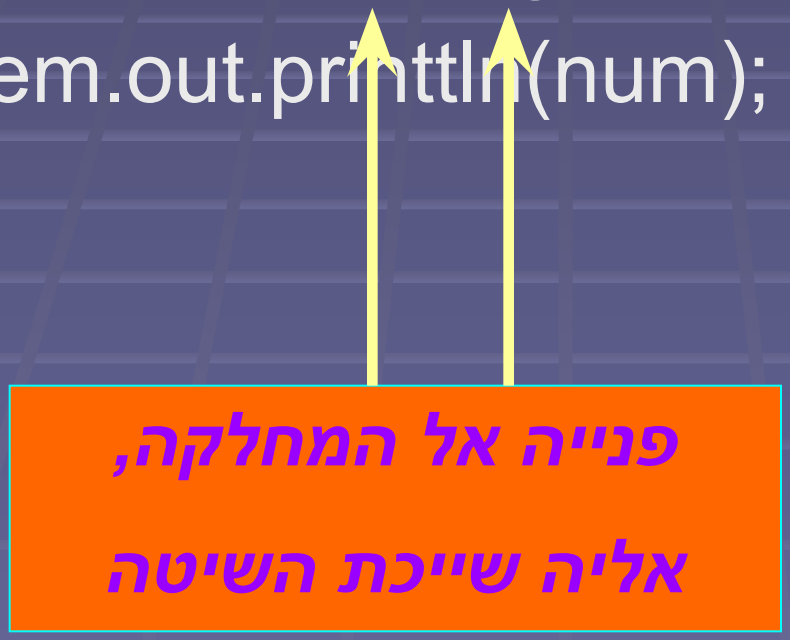
כמה תלמידים נוצרו ?

```
public static void main(String[] args) {
    int num = Student.getCoutner();
    System.out.printtln(num);
}
```

ש בשיטות מחלקה

כמה תלמידים נוצרו ?

```
public static void main(String[] args) {
    int num = Student.getCoutner();
    System.out.printtln(num);
}
```



Pro’s and

- המשתנה הסטטי חוסך זכרון
- מאפשר קשר בין עצמים
- מאפשר ניהול עצמי המחלקה
- מבצע בפשטות פעולה שהן מסובכות בלעדיו.

- שובר את ההגיון של עצם ומחלקה סגורים.
- פתח להשפעות הדדיות של עצמים זה על זה.
- פתח לטעויות.

עים בשימוש המחלקה

- אין טעם ליצור משתנה קבוע עבור כל עצם.
- כמו כן, בגלל שהוא קבוע, אין חשש שישונה.
- לכן, נהוג ליצור קבועים של מחלקה באופן הבא:

```
public static double GRAVITY = 9.81;
```

- לקבועים אלה ניתן לגשת גם מחוץ למחלקה:

```
double newton =
        avi.getMass() * Student.GRAVITY;
```

זאת בהנחה שהגדרנו את הקבוע במחלקה Student

ולות שאינן קשורות לעצם

- יש לא מעט פעולות שאינן קשורות לעצם:
  חישובים מתימטיים
  ריקורסיה

- לשם כך יוצרים 'מחלקות שירות' שכל תפקידן לספק את השירותים האלה.

של – המחלקה MATH

- http://java.sun.com/j2se/1.3/docs/api
- כל התכונות קבועים סטטים (E, PI)
- כל השיטות סטטיות, למשל
- public static int abs(double d)

- חוסך יצירת עצם חסר משמעות.

JAVA – C

- למעשה, שיטות סטטיות זהות לפונקציות ב-C, כי אינו משוייכות לעצם.
- כך גם יוצרים **ריקורסיה** ב-java: שיטה סטטית, הקוראת לעצמה.

- תכונות (איברי) מחלקה : אתחול וגישה
- שיטות מחלקה: הגדרה וגישה
- קבועים סטטים
- מחלקות עזר
- ריקורסיות