

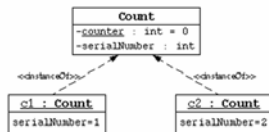
תכונות ברמת המחלקה (static members)

אוהד ברזילי
תכנות מתקדם בשפת Java
אוניברסיטת תל אביב

static keyword

- שדות המוגדרים כ static מציינים כי הם מוגדרים ברמת המחלקה ולא ברמת עצם כל העצמים של אותה מחלקה **חולקים** את אותם נתונים עצמם
- שדות אלו נקראים שדות מחלקה (class members) להבדיל משדות סתם (instance members)

השדה counter הוא סטטי



```
public class Count {  
    private int serialNumber;  
    public static int counter = 0;  
  
    public Count() {  
        counter++;  
        serialNumber = counter;  
    }  
}
```

גישה למשתנה סטטי

- הדבר דומה למשתנים גלובלים בשפות אחרות (תוך שימוש ב namespace)
- אם השדה הסטטי אינו private ניתן לגשת אליו תוך ציון שמו המלא:

```
public class OtherClass {  
    public void incrementNumber() {  
        Count.counter++;  
    }  
}
```

גישה למשתנה סטטי

- אם השדה אינו נגיש, נפנה אליו דרך מתודה של המחלקה שבה הוא מוגדר (getter)
- בעיה: כדי להשתמש במתודה של עצם יש ליצור עצם מאותה מחלקה
- פתרון: נגדיר את מתודת הגישה לנתון הסטטי כסטטית בעצמה
- מתודה כזו נקראת שרות או שרות מחלקה (להבדיל מ instance method)

שרות מחלקה

```
public class Count {  
    private int serialNumber;  
    private static int counter = 0;  
    public static int getTotalCount() {  
        return counter;  
    }  
  
    public Count() {  
        counter++;  
        serialNumber = counter;  
    }  
}
```

שרות מחלקה

- כעת ניתן לדעת את מספר המופעים גם מחוץ למחלקה:

```
public class TestCounter {
    public static void main(String[] args) {
        System.out.println("Number of counter is "
            + Count.getTotalCount());
        Count count1 = new Count();
        System.out.println("Number of counter is "
            + Count.getTotalCount());
    }
}
```

- מתודות רגילות יכולות לגשת לנתונים סטטיים (לא תמיד נוח) אבל להיפך לא – מתודות סטטיות אינן יכולות לגשת לשדות מופע (מדוע?)

שרות מחלקה

```
public class Count {
    private int serialNumber;
    private static int counter = 0;
    public static int getSerialNumber(){
        return serialNumber; // COMPILER ERROR!
    }
}
```

- המשתנה serialNumber מוגדר בכל עצם מהמחלקה Count. לא ברור לאיזה עצם לגשת.
- יתר על כן, אולי המתודה getSerialNumber תיקרא לפני שהוגדרו עצמים מהמחלקה Count

אתחול סטטי

- שדות "רגילים" מאותחלים בבנאי. איפה יאותחלו שדות המחלקה? (ומתי?)
- האתחול יכול להיות טריוויאלי (למשל ערך ברירת מחדל) אך גם יכול לגרום חישוב כלשהו בבנאי? מאוחר מדי
- יש צורך להגדיר "בנאי סטטי" – קטע קוד שירוצ מיד כאשר ה JVM מגלה שהוא עומד להשתמש במחלקה והוא טוען אותה לזיכרון (class loader)
- קטע זה נקרא בלוק סטטי (static block)

אתחול סטטי

- לדוגמא: נרצה לאתחל את המשתנה counter לפי ערך משתנה (בשם myApp.Count.counter) שיוזן ע"י המשתמש בשורת הפקודה

```
public class Count {
    public static int counter;
    static {
        counter = Integer.getInteger("myApp.Count.counter").intValue();
    }
}
```

```
הרצת התוכנית:
> java -DmyApp.Count.counter=47 TestStaticInit
counter = 47
```

```
public class TestStaticInit {
    public static void main(String[] args) {
        System.out.println("counter = "+ Count.counter);
    }
}
```

קבועים

- מקובל להגדיר ב Java קבועי מחלקה ע"י המציינים: static final
- לדוגמא:

```
// Math.class
public static final double PI = 3.14159...

// Bank.class
private static final double DEFAULT_INTEREST_RATE=3.2;
```

enums

- כמו כן היה מקובל להגדיר כך גם enums (טיפוסים שכל הליטרלים שלהם ידועים מראש):

```
public static final int WINTER = 1;
public static final int SPRING = 2;
public static final int SUMMER = 3;
public static final int AUTUM = 4;
```

- החל ב Java5 הוסף הטיפוס enum לשפה והוא מחליף מוסכמה זו (יילמד בהמשך הקורס)

Singleton Design Pattern

תרגיל בית:

- איך בעזרת שימוש במשתנים ובתכונות סטטיות של מחלקה ניתן לאכוף יצירה של עצם ממחלקה מסוימת לכל היותר פעם אחת בתוכנית?