



עצמים

תכנות מתקדם בשפת Java

אוהד ברזילי

אוניברסיטת תל אביב



עצמים

המצגת מכילה קטעים מתוך מצגת של פרופ' עמירם יהודאי
ע"פ הספר:

Object-Oriented Software Construction, 2nd edition,
by Bertrand Meyer (Prentice Hall) .

כל הזכויות שמורות למחברים



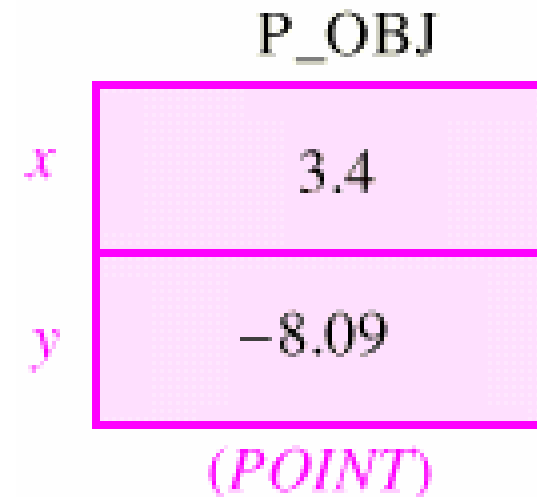
עצמים

- עצם הוא מופע זמן-ריצה (instance) של מחלקה
- עצם הוא מופע ישיר של בדיוק מחלקה אחת, המחלקה היוצרת, ואולי מופע של מחלקות נוספות כתוצאה מירושה
- מבנה הנתונים המייצג עצם מכיל שדות נתונים בלבד ללא פונקציות (מתודות)



POINT Class

```
public class POINT {  
    private float x, y;  
    // Routines...  
}
```



Simple Book

```
public class BOOK1 {  
    private String title;  
    private int date, page_count;  
}
```

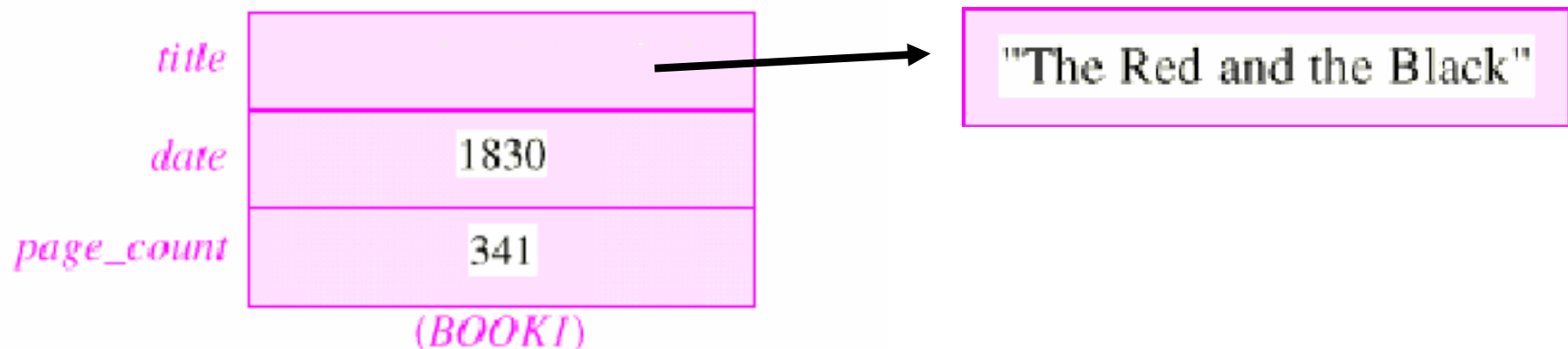
<i>title</i>	"The Red and the Black"
<i>date</i>	1830
<i>page_count</i>	341

(BOOK1)

התרשים פשטני –
מחרוזת היא עצם ולכן
השדה title מכיל רק
הפנייה אליו

Simple Book

```
public class BOOK1 {  
    private String title;  
    private int date, page_count;  
}
```





Writer Class

```
public class WRITER {  
    private String name, real_name;  
    private int birth_year, death_year;  
}
```

<i>name</i>	"Stendhal"
<i>real_name</i>	"Henri Beyle"
<i>birth_year</i>	1783
<i>death_year</i>	1842

(*WRITER*)



עצמים

- נסווג את השדות לסוגים לפי טיפוסיהם:
 - טיפוסים יסודיים (פרימיטיבים): חלק משפת התכנות. כגון: `int`, `char`, `float`, `bool` (אבל לא `string`)
 - טיפוסים מורכבים (user defined types)
- שדה מורכב: עצם מוכל או עצם מוצבע?

תת עצמים

<i>title</i>	"The Red and the Black"								
<i>date</i>	1830								
<i>page_count</i>	341								
	<table><tr><td><i>name</i></td><td>"Stendhal"</td></tr><tr><td><i>real_name</i></td><td>"Henri Beyle"</td></tr><tr><td><i>birth_year</i></td><td>1783</td></tr><tr><td><i>death_year</i></td><td>1842</td></tr></table> <p>(WRITER)</p>	<i>name</i>	"Stendhal"	<i>real_name</i>	"Henri Beyle"	<i>birth_year</i>	1783	<i>death_year</i>	1842
<i>name</i>	"Stendhal"								
<i>real_name</i>	"Henri Beyle"								
<i>birth_year</i>	1783								
<i>death_year</i>	1842								

(BOOK2)

<i>title</i>	"Life of Rossini"								
<i>date</i>	1823								
<i>page_count</i>	307								
	<table><tr><td><i>name</i></td><td>"Stendhal"</td></tr><tr><td><i>real_name</i></td><td>"Henri Beyle"</td></tr><tr><td><i>birth_year</i></td><td>1783</td></tr><tr><td><i>death_year</i></td><td>1842</td></tr></table> <p>(WRITER)</p>	<i>name</i>	"Stendhal"	<i>real_name</i>	"Henri Beyle"	<i>birth_year</i>	1783	<i>death_year</i>	1842
<i>name</i>	"Stendhal"								
<i>real_name</i>	"Henri Beyle"								
<i>birth_year</i>	1783								
<i>death_year</i>	1842								

(BOOK2)

עצמים, מצביעים והתייחסויות

```
Point p1 = new Point();
```

```
Point p2;
```

```
p2 = p1;
```

```
p1.setX(4);
```

x	0
y	0

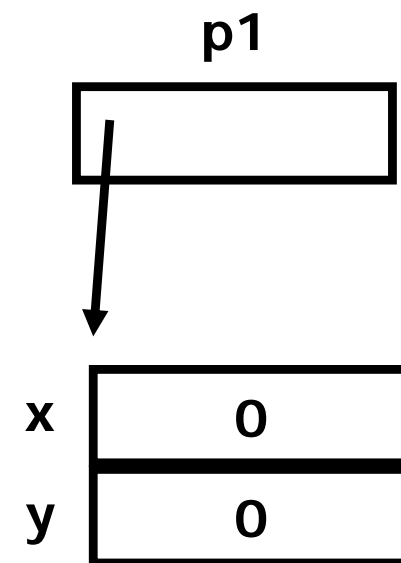
עצמים, מצביעים והתייחסויות

```
Point p1 = new Point();
```

```
Point p2;
```

```
p2 = p1;
```

```
p1.setX(4);
```



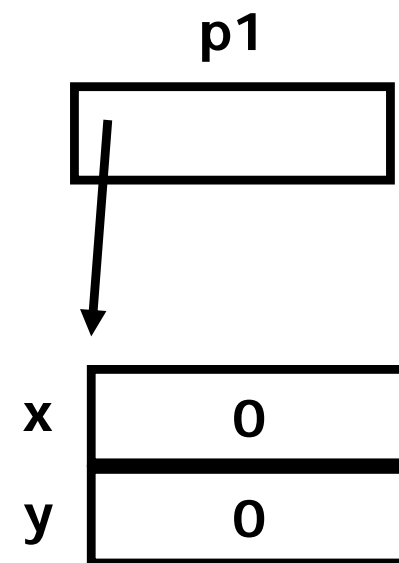
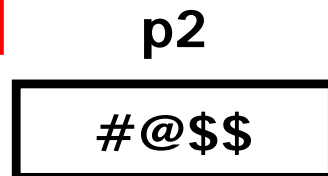
עצמים, מצביעים והתייחסויות

```
Point p1 = new Point();
```

```
Point p2;
```

```
p2 = p1;
```

```
p1.setX(4);
```



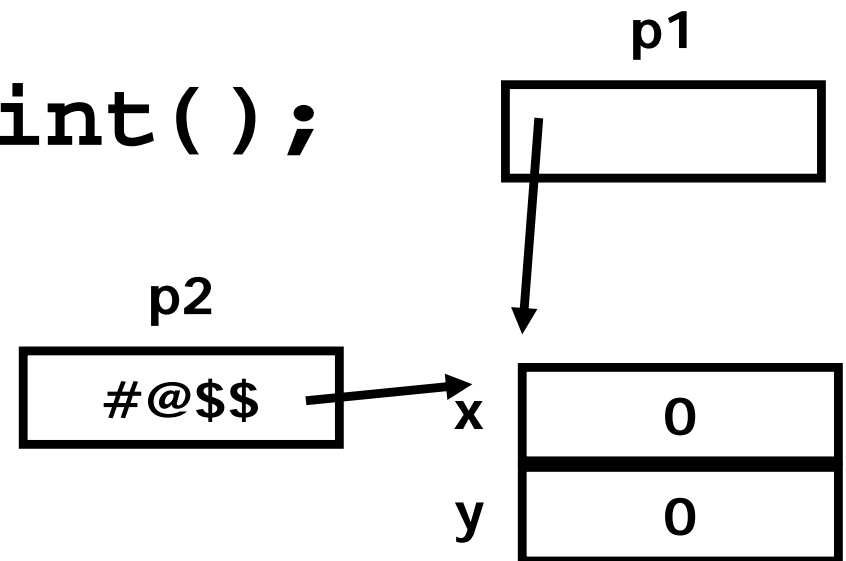
עצמים, מצביעים והתייחסויות

```
Point p1 = new Point();
```

```
Point p2;
```

```
p2 = p1;
```

```
p1.setX(4);
```



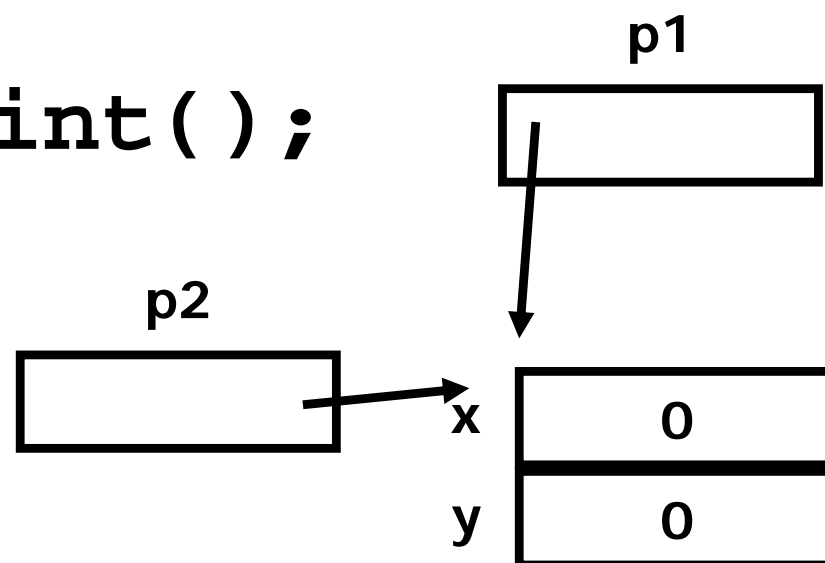
עצמים, מצביעים והתייחסויות

```
Point p1 = new Point();
```

```
Point p2;
```

```
p2 = p1;
```

```
p1.setX(4);
```



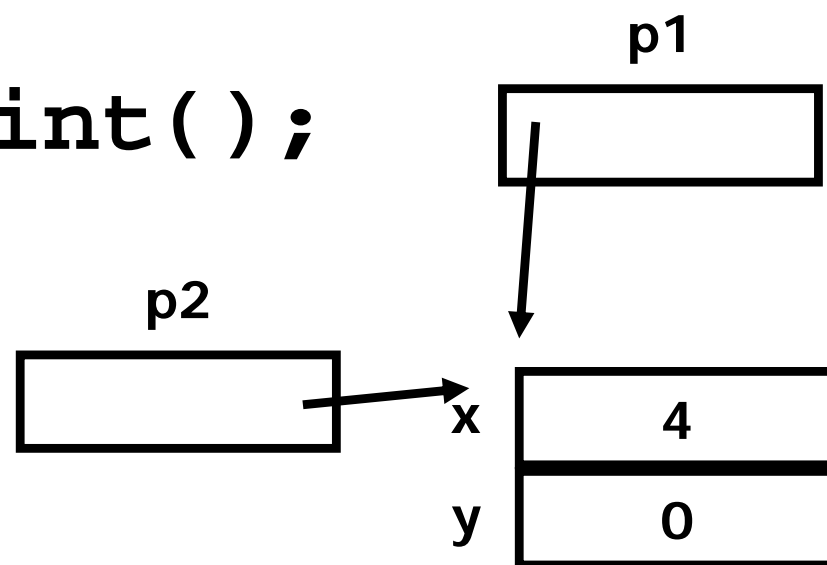
עצמים, מצביעים והתייחסויות

```
Point p1 = new Point();
```

```
Point p2;
```

```
p2 = p1;
```

```
p1.setX(4);
```





עצמים, מצביעים והתייחסויות

- נשים לב כי כל העצמים בשפת Java הם אנונימיים (חסרי שם)
 - להבדלי משפות אחרות (כגון ++C)
- הפנייה ב Java היא ישות עצמאית, משתנה בפני עצמו, שעשויה להכיל כתובת של אובייקט קיים. ניתן לעדכן את ערכה של ההפנייה ובכך להחליף את העצם המוצבע.
- לא נתבלבל עם העברת נתונים לפונקציה by reference. העתקה או אי העתקת העצם לא תלויה בעצם עצמו אלא בהגדרת שיטת ההעברה בפונקציה

עצם מוכל או מוצבע?

- תת עצמים תופסים יותר זכרון
- עצמים מוצבעים קל יותר לשתף בין מספר ישויות
- לא כל המצביעים מאותחלים אוטומטית (ל- null) כך שקיימות 3 אפשרויות:

- המצביע מתייחס לעצם אחד מסוים
- מצביע לכתובת 'זבל'

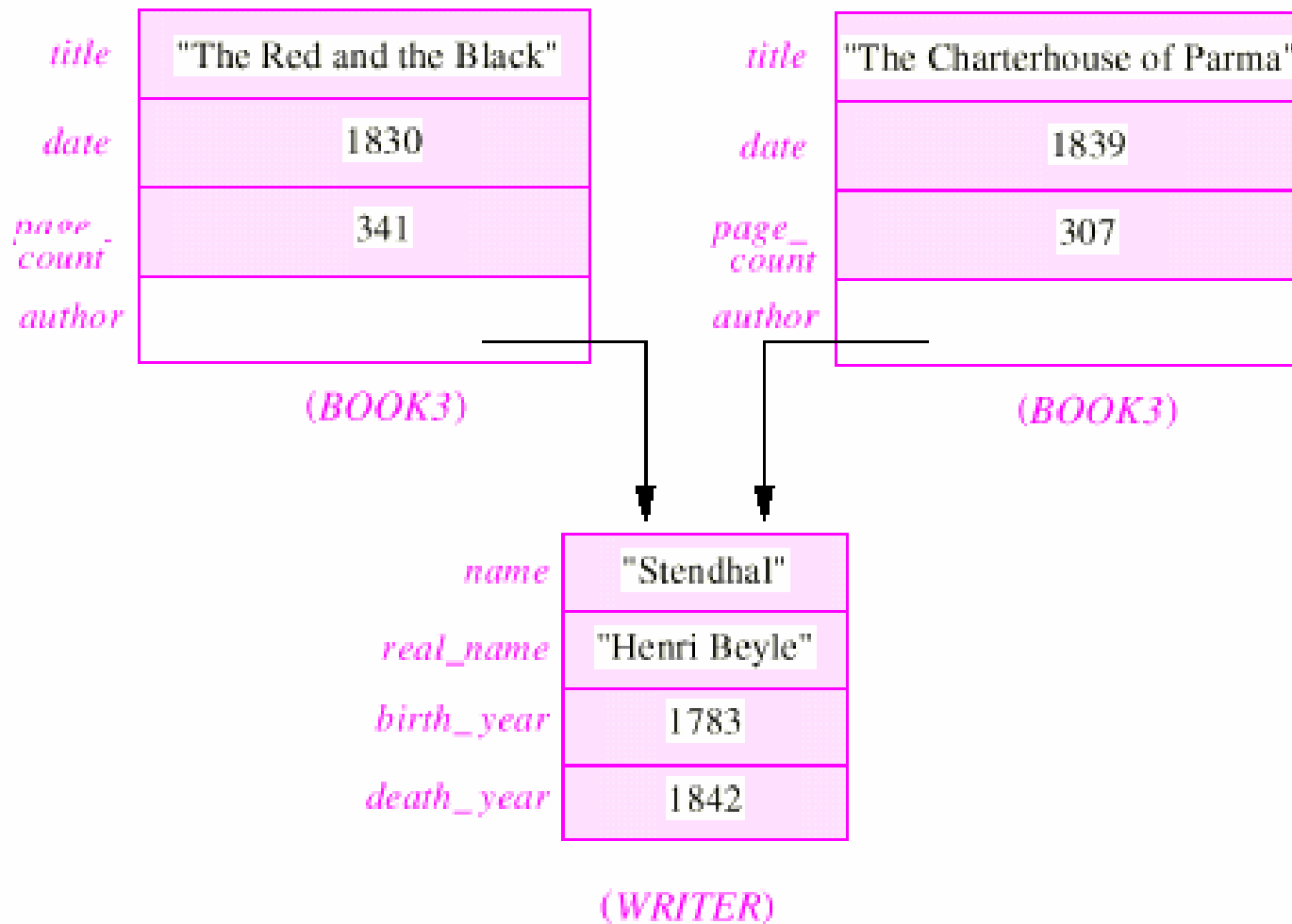
- המצביע הוא null כלומר לא משויך לעצם מסוים

- בשפת Java השאלה לא קיימת מכיוון שאין עצמים מוכלים

title	"The Red and the Black"
date	1830
page count	341
name	"Stendhal"
real_name	"Henri Beyle"
birth_year	1783
death_year	1842
	(WRITER)
	(BOOK2)

title	"Life of Rossini"
date	1823
page count	307
name	"Stendhal"
real_name	"Henri Beyle"
birth_year	1783
death_year	1842
	(WRITER)
	(BOOK2)

מוצבע



לא משויך לאף עצם

<i>title</i>	"Candide, or Optimism"
<i>date</i>	1759
<i>page count</i>	120
<i>author</i>	

(BOOK3)



זהות של עצמים

- שני עצמים נפרדים עשויים להיות זהים
- שוויון שדות אינו תנאי מספיק לזהותם של שני עצמים

```
class WRITER {  
    String name;  
    int birth_year, death_year;  
}
```

```
class BOOK {  
    String title;  
    int date;  
    WRITER *author;  
}
```

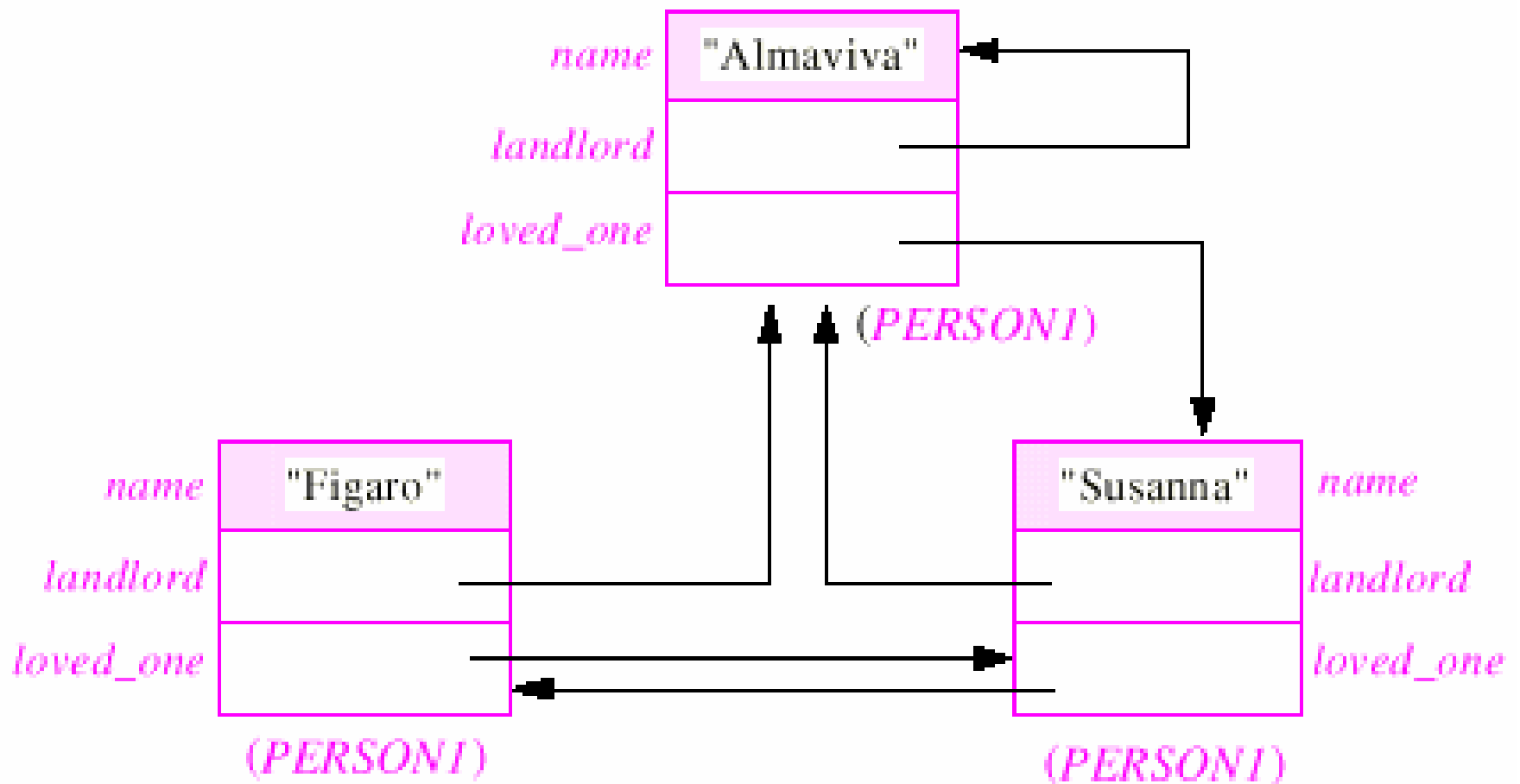


מעגלים ביחס הלקוחות

- מחלקה עשויה להיות לקוחה של עצמה
- נבדיל בין מעגל של עצמים ומעגל של מחלקות

```
class PERSON1{  
    string name;  
    PERSON1 loved_one, landlord;  
}
```

מצביע לעצמו





יתרונות העצם המוכל

■ יעילות

- גישה לשדות מוכלים שלא דרך dereference של מצביע

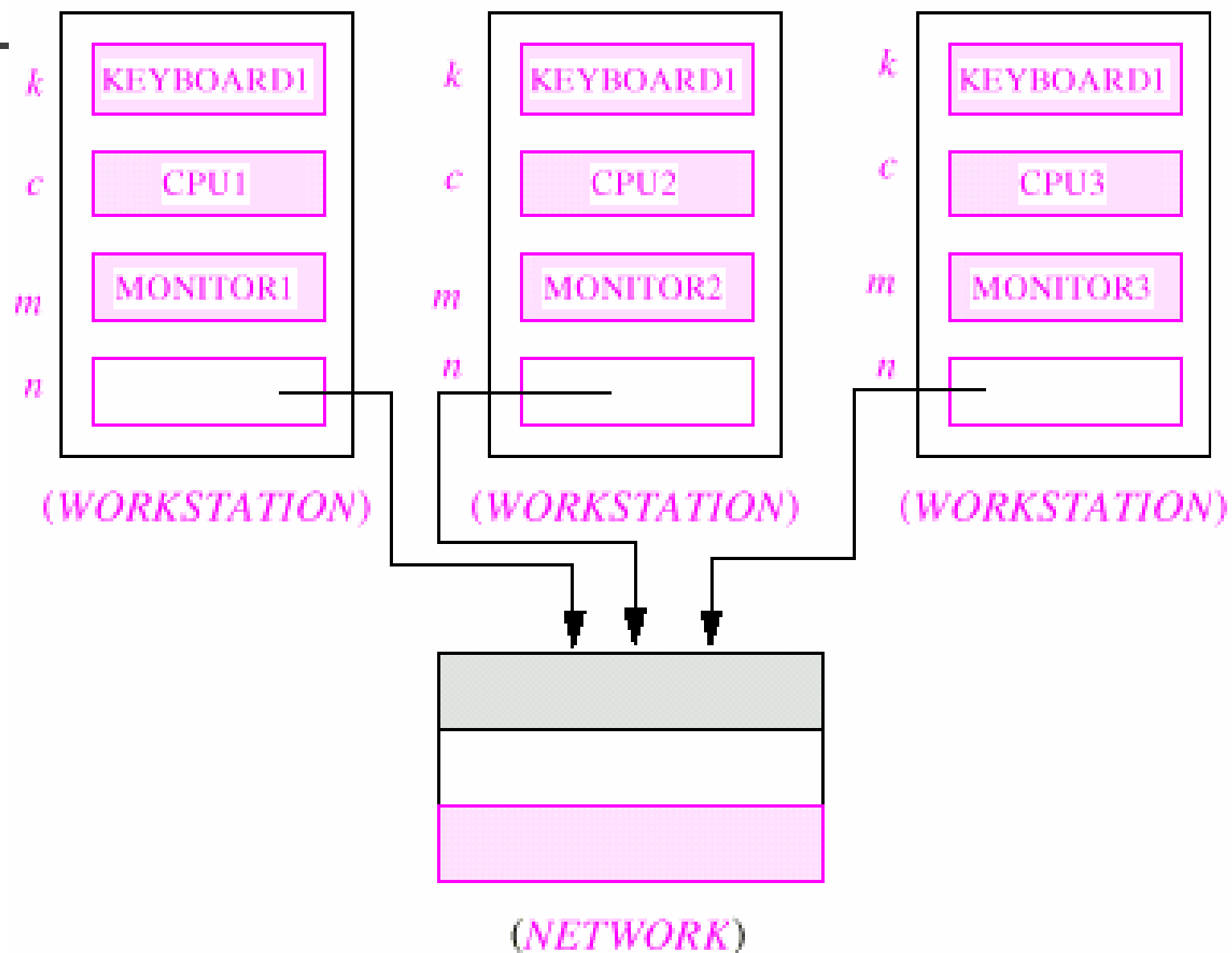
■ מודל טוב יותר

- מצביע למחלקה S פירושו שהלקוח "יודע על" S
- עצם כעצם מוכל מעיד על כך שהלקוח מכיל S
- בפרט, הכלה מרמזת על אי-שיתוף

■ תמיכה אחידה בטיפוסים פרימיטיביים

- עצמים מכילים את הטיפוסים היסודיים עצמם ולא מצביע אליהם

אובייקטים מורכבים





עצמים מוכלים

- לא יתכנו מעגלי ספק-לקוח כאשר הספק הוא עצם מוכל (expanded type)
- יצירה והריסה של האובייקט המוכל תלויה בזמני היצירה וההריסה של העצם המכיל
- בהשמת העצם העוטף לא יופעל בהכרח אופרטור ההשמה של כל אחד משדותיו המוכלים
- כאשר עצם מכיל מצביעים לעצמים אחרים יש לשקול מתי לשחרר את הזכרון של עצמים אלו (למשל אם הם משותפים לכמה עצמים)



הכלה או הצבעה?

- בשפת Java הוחלט **שלא לאפשר הכלת עצמים**

- כל ההתייחסויות לעצמים הן הפניות



פונקציה יוצרת

- יצירת ואתחול עצמים מפורשת
- כאשר בפונקציה כלשהי מגיעים לשורה
 $x = \text{new } T()$ קורים הדברים הבאים:
 - .1 מוקצה מקום עבור עצם מטיפוס T (עבור כל אחד משדותיו)
 - .2 x מקבל את כתובתו של העצם שהוקצה



מצביעים ומצבם

- מצביע יכול להיות קשור או לא קשור לעצם
- מצביע p יקשר לעצם e :
 - $p = \text{new SomeClass}(\dots)$
 - או e יי $p = p1$, כאשר $p1$ הוא מצביע קשור
- p יהפך ללא קשור e :
 - $p = \text{null}$
 - או e יי $p = p1$ כאשר $p1$ הוא null
- הפעלת מתודה דרך מצביע שאינו קשור תגרום לטעות זמן ריצה



פעולות על מצביעים

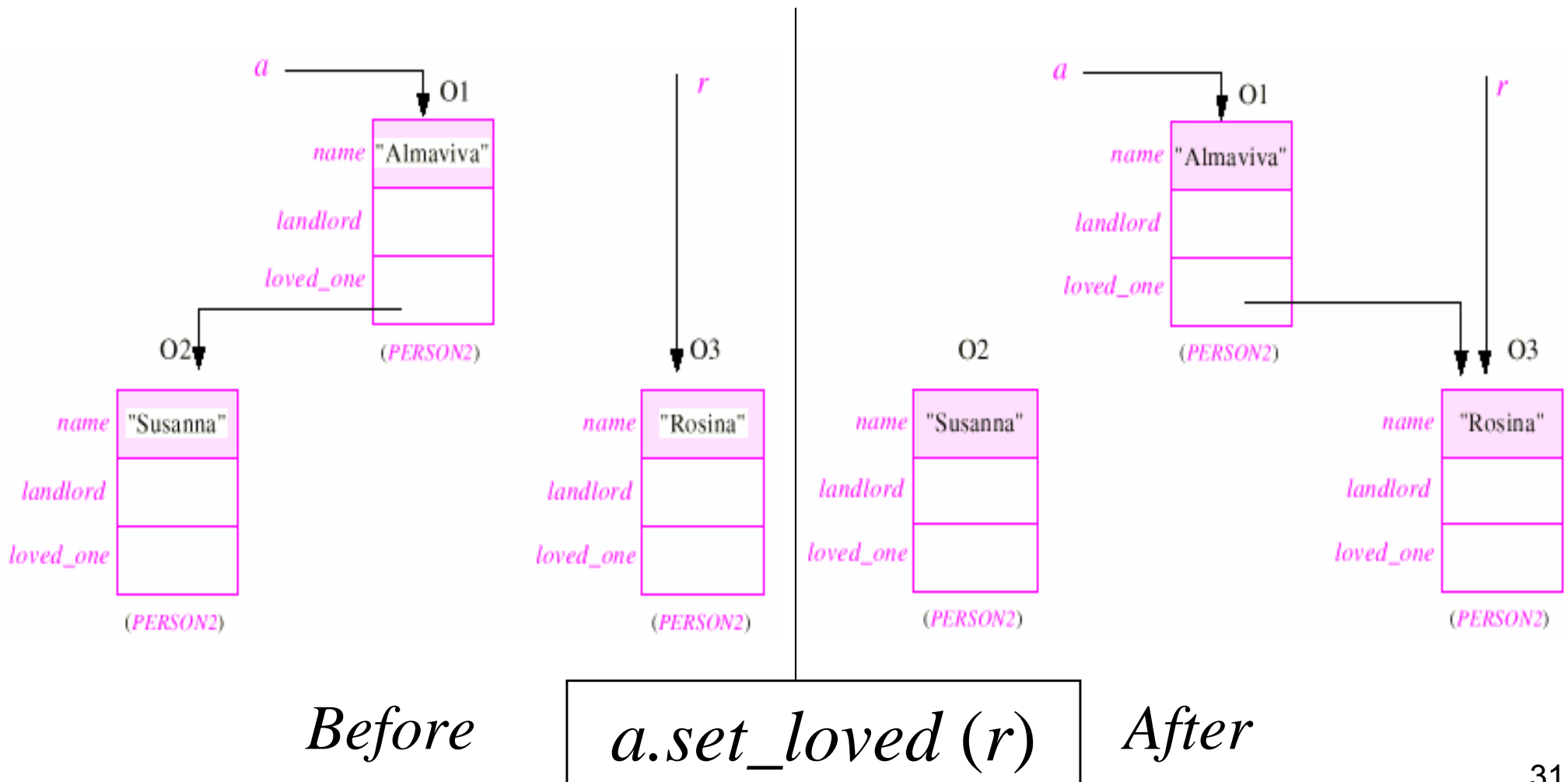
- השמה של מצביעים אינה מבצעת העתקה של האובייקט המוצבע
- ההשוואה בין מצביעים $p1 == p2$ תחזיר true רק אם הם מצביעים על אותו עצם עצמו או ששניהם null
- ההשמה $p = \text{null}$ שימושית כדי:
 - לבדוק בהמשך האם $p \neq \text{null}$ (שכן למצביע עלול להיות גם ערך 'זיבלי')
 - לעזור ל Garbage Collector להבין שהעצם המוצבע אינו נחוץ עוד



קשירת מצביע

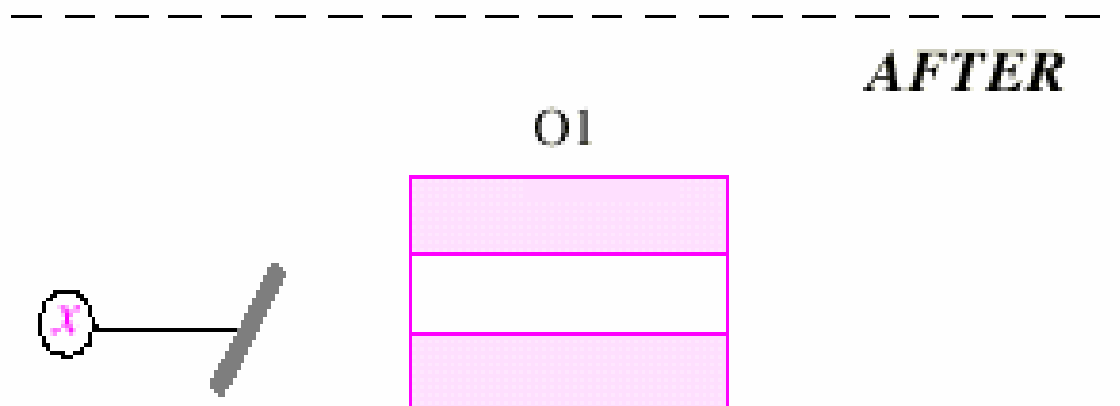
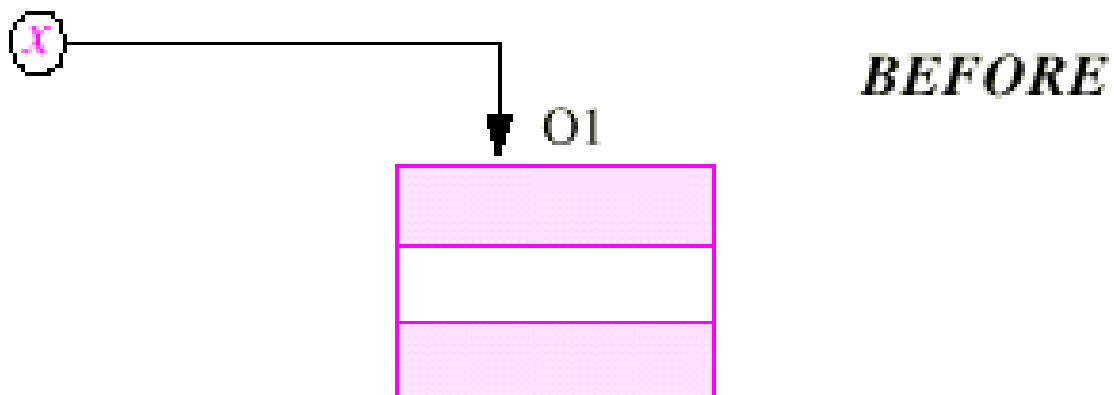
```
class PERSON2 {  
    String name;  
    PERSON2 loved_one, landlord;  
  
    /**Attach the loved_one field of  
    current object to l. */  
    void set_loved (PERSON2 l){  
        loved_one = l;  
    }  
}
```

קשירת מצביע



איפוס מצביע (ללא שחרור)!

$x = null$





פונקציה יוצרת

- במחלקות שנכתוב ניתן (ורצוי?) להחליף את אתחול ברירת המחדל בפונקציה
- פונקציה זו נקראת הבנאי (constructor) של המחלקה
- ניתן להעמיס אותה
- ניתן להעביר לה פרמטרים
- שמה כשם המחלקה שאותה היא מאתחלת ואין לה ערך מוחזר



בנאים ל Point

```
public class Point {  
    //...  
    public Point() {  
        m_x = 0;  
        m_y = 0;  
    }  
  
    public Point(int x, int y) {  
        m_x = x;  
        m_y = y;  
    }  
    //...  
}
```



ריבוי בנאים

- כדי לשמור על עיקביות בתחזוקת מספר גרסאות מועמסות של אותה הפונקציה מומלץ להשתמש בקריאות הדדיות בין הגרסאות המועמסות

- בהעמסת בנאים, הקריאה `this(args)` קוראת לבנאי מועמס שחתימתו תואמת את `args`



בנאי יותר טוב ל Point

```
public class Point {  
    //...  
    public Point(int x, int y) {  
        m_x = x;  
        m_y = y;  
    }  
    public Point() {  
        this(0,0);  
    }  
    //...  
}
```



עוד על בנאים

- פונקציות אתחול שהוגדרה כ `private` לא תאפשר ליצור באמצעותה עצמים
- למחלקה שלא נכתבו עבורה בנאים כלל מספק המהדר (הקומפילר) בנאי ברירת מחדל (`default constructor`) זהו בנאי חסר פרמטרים שאינו עושה דבר
- מרגע שנכתב בנאי כלשהו למחלקה לא ניתן עוד להשתמש בבנאי ברירת המחדל
- יש מקרים שבהם נהיה חייבים להשתמש בבנאי חסר פרמטרים ולכן נגדיר כזה לכל מחלקה



יצירה ואתחול מפורשים

- עצמים נוצרים רק ע"י שימוש מפורש באופרטור `new`

- 2 יוצאי דופן:

- אתחול מערך:

```
int [] arr = {1, 2, 3};
```

- יצירת מחרוזות ע"י מרכאות:

```
String hello = "hello";
```