

תוכנה 1

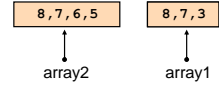
תרגול 5: שאר ירקות
רובי בוים ומתי שמרת

דרכים נוספות להעתקה

הפונקציה `arraycopy` במחלקה `java.lang.System` מאפשרת העתקת תוכנו של מערך אחד לאחר

```
public static void arraycopy(Object src, int srcPos,
                             Object dest, int destPos,
                             int length)
System.arraycopy(array2, 0, array1, 0, 2);
```

1,2 in array1 are replaced with 8,7



Details:

<http://java.sun.com/javase/6/docs/api/java/lang/System.html>

המרת טיפוסים פרימיטיביים

הצרה Narrowing

- short to byte or char
- char to byte or short
- int to byte, short, or char
- long to byte, short, char, or int
- float to byte, short, char, int, or long
- double to byte, short, char, int, long, or float

הרחבה Widening

- byte to short, int, long, float, or double
- short to int, long, float, or double
- char to int, long, float, or double
- int to long, float, or double
- long to float or double
- float to double

3

הרחבה

- מתבצעת באופן אוטומטי ע"י הקומפילר
- הרחבה מטיפוס אינטגרלי אחד לאחר וכן מ float ל double היא מדייקת
- בהרחבה של int או long ל float או של long ל double ייתכן אובדן דיוק.

```
public static void wideningExample() {
    int ibig = 1234567890;
    float fapprox = ibig;
    System.out.println(ibig - (int) fapprox); // -46
    long lbig = 998877665544332211L;
    double dapprox = lbig;
    System.out.println(lbig - (long) dapprox); // 51
}
```

4

הצרה

- דורשת המרה מפורשת (casting) ע"י המתכנת
- ייתכן אובדן של מידע

```
private static void wideningExample() {
    float fmin = Float.NEGATIVE_INFINITY;
    float fmax = Float.POSITIVE_INFINITY;
    long l = -9223372036854775808L..9223372036854775807L;
    println("long: " + (long) fmin + "..." + (long) fmax);
    println("int: " + (int) fmin + "..." + (int) fmax);
    println("short: " + (short) fmin + "..." + (short) fmax);
    println("char: " + (int) (char) fmin + "..." + (int) (char) fmax);
    println("byte: " + (byte) fmin + "..." + (byte) fmax);
}
```

5

הצרה

- דורשת המרה מפורשת (casting) ע"י המתכנת
- ייתכן אובדן של מידע

```
private static void wideningExample() {
    float fmin = Float.NEGATIVE_INFINITY;
    float fmax = Float.POSITIVE_INFINITY;
    int i = -2147483648..2147483647;
    println("int: " + (int) fmin + "..." + (int) fmax);
    println("short: " + (short) fmin + "..." + (short) fmax);
    println("char: " + (int) (char) fmin + "..." + (int) (char) fmax);
    println("byte: " + (byte) fmin + "..." + (byte) fmax);
}
```

6

הצרה

- דורשת המרה מפורשת (casting) ע"י המתכנת
- ייתכן אובדן של מידע

```
private static void wideningExample() {
    float fmin = Float.NEGATIVE_INFINITY;
    float fmax = Float.POSITIVE_INFINITY;

    println("long: " + (long) fmin + " " + (long) fmax);
    println("short: 0..-1");
    println("short: " + (short) fmin + " " + (short) fmax);
    println("char: " + (int) (char) fmin + " " + (int) (char) fmax);
    println("byte: " + (byte) fmin + " " + (byte) fmax);
}
```

7

הצרה

- דורשת המרה מפורשת (casting) ע"י המתכנת
- ייתכן אובדן של מידע

```
private static void wideningExample() {
    float fmin = Float.NEGATIVE_INFINITY;
    float fmax = Float.POSITIVE_INFINITY;

    println("long: " + (long) fmin + " " + (long) fmax);
    println("int: " + (int) fmin + " " + (int) fmax);
    println("short: " + (short) fmin + " " + (short) fmax);
    println("char: " + (int) (char) fmin + " " + (int) (char) fmax);
    println("byte: " + (byte) fmin + " " + (byte) fmax);
}
```

8

הצרה

- דורשת המרה מפורשת (casting) ע"י המתכנת
- ייתכן אובדן של מידע

```
private static void wideningExample() {
    float fmin = Float.NEGATIVE_INFINITY;
    float fmax = Float.POSITIVE_INFINITY;

    println("long: " + (long) fmin + " " + (long) fmax);
    println("int: " + (int) fmin + " " + (int) fmax);
    println("short: " + (short) fmin + " " + (short) fmax);
    println("char: " + (int) (char) fmin + " " + (int) (char) fmax);
    println("byte: " + (byte) fmin + " " + (byte) fmax);
}
```

9

Numeric Promotion

- הרחבה של האופרנדים בביטוי מתימטי
- טיפוס משותף לשני האופרנדים לצורך ביצוע הפעולה
- אם אחד האופרנדים הוא double גם השני יומר ל double
- אם אחד האופרנדים הוא float גם השני יומר ל float
- אם אחד האופרנדים הוא long גם השני יומר ל long
- שני האופרנדים יומר ל int

10

עוד דוגמאות

```
public static void main(String[] args) {
    long l = 2000000000+20000000000; //l == -294967296
    int i = (int) 1.99999999; //i == 1
    float f = (float) 1.99999999; //f == 2
    f = 5/2; //f == 2
    f = (float) (5/2); //f == 2
    f = (float) 5/2; //f == 2.5
    f = 5 / (float) 2; //f == 2.5
    short a = 2; // why is this ok?
    short c = a*a; // compilation error: cannot convert int to short
}
```

עוד על המרות ב-

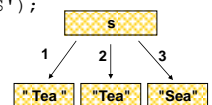
http://java.sun.com/docs/books/jls/third_edition/html/conversion_s.html

11

מחרוזות

- מרגע שנוצרה המחרוזת היא אינה ניתנת לשינוי (immutable)
- ההפניה למחרוזת כמובן יכולה להשתנות

```
String s = " Tea ";
s = s.trim();
s = s.replace('T', 'S');
```

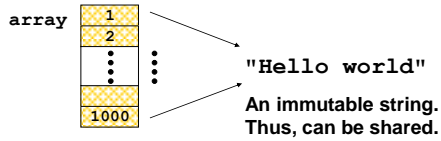


12

Interning

■ מכיוון שמחזרות הן קבועות ניתן לשתף אותן

```
String[] array = new String[1000];
for (int i = 0; i < array.length; i++) {
    array[i] = "Hello world ";
}
```



13

דוגמא ל-Interning

```
String hello = "Hello", lo = "lo";  
System.out.println(hello == "Hello");  
System.out.println(Other.hello == hello);  
System.out.println(hello == ("Hel"+"lo"));  
System.out.println(hello == ("Hel"+lo));  
System.out.println(hello == ("Hel"+lo).intern());
```

String literals

14

דוגמא ל-Interning

```
String hello = "Hello", lo = "lo";  
System.out.println(hello == "Hello");  
System.out.println(hello == ("Hel"+"lo"));  
System.out.println(hello == ("Hel"+lo));  
System.out.println(hello == ("Hel"+lo).intern());
```

System.out.println(hello == "Hello");
System.out.println(hello == ("Hel"+"lo"));

Literal strings within the same class represent references to the same String

15

דוגמא ל-Interning

```
String hello = "Hello", lo = "lo";  
System.out.println(hello == "Hello");  
System.out.println(Other.hello == hello);  
System.out.println(hello == ("Hel"+"lo"));  
System.out.println(hello == ("Hel"+lo).intern());
```

System.out.println(Other.hello == hello);
System.out.println(hello == ("Hel"+"lo"));

Literal strings within different classes represent references to the same String object

16

דוגמא ל-Interning

```
String hello = "Hello", lo = "lo";  
System.out.println(hello == "Hello");  
System.out.println(Other.hello == hello);  
System.out.println(hello == ("Hel"+"lo"));  
System.out.println(hello == ("Hel"+lo).intern());
```

System.out.println(hello == ("Hel"+"lo"));

Strings computed by constant expressions are computed at compile time and then treated as if they were literals

17

דוגמא ל-Interning

```
String hello = "Hello", lo = "lo";  
System.out.println(hello == "Hello");  
System.out.println(Other.hello == hello);  
System.out.println(hello == ("Hel"+"lo"));  
System.out.println(hello == ("Hel"+lo));  
System.out.println(hello == ("Hel"+lo).intern());
```

System.out.println(hello == ("Hel"+"lo"));
System.out.println(hello == ("Hel"+lo));

Strings computed by concatenation at run time are newly created and therefore distinct

18

דוגמא ל-Interning

```
String hello = "Hello", lo = "lo";  
  
System.out.println(hello == "Hello");  
  
System.out.println(Other.hello == hello);  
  
System.out.println(hello == ("Hel"+"lo"));  
  
System.out.println(hello == ("Hel"+lo));  
  
System.out.println(hello == ("Hel"+lo).intern());
```

Explicitly interning a String returns a reference to the interned String object. If such a String was previously interned the returned value will refer to that object

19

static field

- שדה המוגדר static (class variable) אינו משייך לאובייקט מסוים
- קיימת בדיוק התגלמות אחת שלו
- אם שדה אינו מוגדר static (instance variable) בכל פעם שנוצר אובייקט חדש נוצר משתנה חדש המשייך לאובייקט זה.

20

static method

- מתודה סטטית (class method) אינה משייכת לאובייקט
- אין אובייקט נוכחי (this)
- מתודה שאינה סטטית (instance method) נקראת תמיד תוך התייחסות לאובייקט מסוים. אובייקט זה הוא האובייקט הנוכחי (this) במתודה
- נוכל לגשת רק לשדות שהוגדרו סטטיים

21

static example

```
class Point {  
    int x, y, useCount;  
    Point(int x, int y) { this.x = x; this.y = y; }  
    static Point origin = new Point(0, 0);  
}  
  
class Test {  
    public static void main(String[] args) {  
        Point p = new Point(1,1);  
        Point q = new Point(2,2);  
  
        p.x = 3; p.y = 3; p.useCount++;  
        p.origin.useCount++;  
  
        System.out.println("(" + q.x + ", " + q.y + "  
        System.out.println(q.useCount);  
        System.out.println(q.origin == Point.origin);  
        System.out.println(q.origin.useCount);  
    }  
}
```

גישה לשדה סטטי יש להשתמש בשם מחלקה ולא באובייקט

true - גישה לאותה הפניה

22

final fields

- ניתן לבצע השמה יחידה למשתנה המוגדר final
- נאכף על ידי הקומפיילר
- שדה סטטי יאותחל יחד עם ההגדרה*
- שדה מופע חייב להיות מאותחל עד סיום כל אחד מבנאי המחלקה
- יחד עם ההגדרה או בבנאי
- שדה שהוגדר final אבל לא אותחל עם הגדרתו נקרא blank final

23