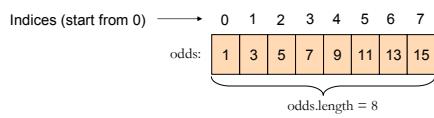


Arrays

- **Array:** A fixed-length data structure for storing multiple values of the same type
- Example: An array of odd numbers:



The type of all elements is int
The value of the element at index 4 is 9: odds[4] == 9

Oranit Dror

2

Software 1

Recitation No. 2 Arrays and Control Structures

Oranit Dror

1

Array Creation and Initialization

- What is the output of the following code:

```
int[] odds = new int[8];
for (int i=0 ; i < odds.length ; i++) {
    System.out.print(odds[i] + " ");
    odds[i] = 2*i+1;
    System.out.print(odds[i] + " ");
}
```

- Output: ?

Oranit Dror

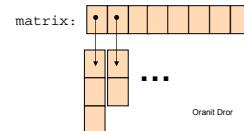
4

Array Declaration

- An array is denoted by the [] notation

- Examples:

- `int[] odds;`
- `int odds[];` // legal but discouraged
- `String[] names;`
- `int[][] matrix;` // an array of arrays



Oranit Dror

3

Loop through Arrays

- By promoting the array's index:

```
for (int i=0 ; i < months.length ; i++) {
    System.out.println(months[i]);
}
```

The variable month is assigned
the next element in each iteration

- In Java 5.0:

```
for (String month: months) {
    System.out.println(month);
}
```

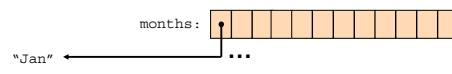
Oranit Dror

6

Array Creation and Initialization

- Creating and initializing small arrays with *a-priori* known values:

- `int[] odds = {1,3,5,7,9,11,13,15};`
- `String[] months = {"Jan", "Feb", "Mar", "Apr", "May", "Jun", "July", "Aug", "Sep", "Oct", "Nov", "Dec"};`



Oranit Dror

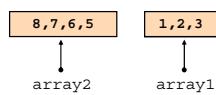
5

Copying Arrays

In the java.lang.System class:

```
public static void arraycopy(Object src, int srcPos,  
                             Object dest, int destPos,  
                             int length)
```

```
System.arraycopy(array2, 0, array1, 0, 2);
```



Details: <http://java.sun.com/j2se/1.5.0/docs/>

Ornit Dror

8

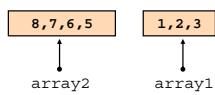
Copying Arrays

Assume:

```
int[] array1 = {1, 2, 3};  
int[] array2 = {8, 7, 6, 5};
```

Naïve copy:

```
array1 = array2;
```



Ornit Dror

7

Other Manipulations on Arrays

The java.util.Arrays class has methods for sorting and searching, assigning arrays e.g.

```
public static void sort(int[] a)  
public static int binarySearch(int[] a, int key)  
public static void fill(long[] a, long val)
```

More details in JDK 5.0 documentation

<http://java.sun.com/j2se/1.5.0/docs/>

Ornit Dror

10

Copying Arrays

What is the output of the following code:

```
int[] odds = {1, 3, 5, 7, 9, 11, 13, 15};  
int[] newOdds = new int[8];  
System.arraycopy(odds, 1, newOdds, 1, 7);  
for (int odd: newOdds) {  
    System.out.print(odd + " ");  
}
```

Output?:

Ornit Dror

9

2D Arrays

```
int[][] table = new int[10][10];  
for (int i = 0 ; i < 10 ; i++) {  
    for (int j = 0 ; j < 10 ; j++) {  
        table[i][j] = (i+1) * (j+1);  
    }  
}
```

Ornit Dror

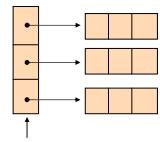
12

2D Arrays

There are no 2D arrays in Java but ...

you can build array of arrays:

```
char[][] board = new char[3][];  
for (int i = 0 ; i < 3 ; i++)  
    board[i] = new char[3];
```



Or equivalently:
`char[][] board = new char[3][3];`

Ornit Dror

11

Switch Statement

```
public class Fibonacci {  
    ...  
  
    /** Returns the n-th Fibonacci element */  
    public static int computeElement(int n) {  
        switch(n) {  
            case 0:  
                return 1;  
            case 1:  
                return 1;  
            break;  
            default:  
                return computeElement(n-1) + computeElement(n-2);  
        }  
    }  
}
```

Granit Pro

10

Switch Statement

```
static int computeElement(int n) {  
    switch(n) {  
        case 0:  
            return 1;  
        case 1:  
            return 1;  
        default:  
            return computeElement(n-1) + computeElement(n-2);  
    }  
}
```

Granit Dror

13

נתונים במקום חישוב

- בתרוגם וקורסיה לולאה אנו משתמשים במשתנה `prevPrev` לשמירת המצביע על `curr`, `prev`, `I` ו- `curr`.
 - הלווא "זכורת" את הנקודה שבה אנו נמצאים בתהליכי החישוב.
 - דיוון: יעילות לעומת פשטות.
 - עיקון-ה-SIMPLE (`KISS`)
 - תרגיל: כתבו את השירות `computeElement` בלבד (ללא `curr`) בעזרת `prevPrev`, `I`, `prev`.

Grenit Pro

10

For Loop

- A loop instead of a recursion

```

static int computeElement(int n) {
    if (n == 0 || n == 1)
        return 1;

    int prev = 1;
    int prevPrev = 1;
    int curr;

    for (int i = 2 ; i < n ; i++) {
        curr = prev + prevPrev;
        prevPrev = prev;
        prev = curr;
    }

    curr = prev + prevPrev;
    return curr;
}

```

Credit Due

16

For Loop

- #### ■ Printing the first n elements:

```
public class Fibonacci {  
    public static int computeElement(int n) {  
        ...  
    }  
  
    public static void main(String[] args) {  
        for(int i = 0 ; i < 10 ; i++)  
            System.out.println(computeElement(i));  
    }  
}
```

Oranit Dre

18

תשובה לתרגיל

```
static int computeElement(int n){  
    if (n==0 || n==1)  
        return 1;  
  
    int prev = 1;  
    int prevPrev = 1;  
  
    for (int i = 2; i < n; i++) {  
        prev = prev + prevPrev;  
        prevPrev = prev - prevPrev;  
    }  
  
    return prev + prevPrev;  
}
```

Oranit Dror

17

מודולריות, שכפול קוד ויעילות

- מетодה (פונקציה) צריכה לעשות דבר אחד בלבד!
- ערוב של חישוב והדפסה פוגע במודולריות (מדוע?)
 - היזהרו משכפול קוד!
- קטוע קוד דומה המופיע בשתי פונקציות שונות יגרום במקדים או לאחר מכן לאגע בתוכנית (מדוע?)
 - את בעיית היעילות (הוספת מגנן memoization)
 - אפשר לפתור בעזרת מעריצים

Ornit Dror

20

מודולריות, שכפול קוד ויעילות

- יש כאן חסור יעילות מסוימת:
 - ללאת ה-for חזרת גם ב- main וגם ב- computeElement ניתן גם לחשב את האברים וגם למופיע אותם
 - כמו כן כדי לחשב איבר בסדרה איןנו משתמשים בתוצאות שכבר חישבנו (של אברים קודמים)
 - ומתחילה כל חישוב מתחילה

Ornit Dror

19

while vs. do while

- The following two statements are equivalent:

```
int i=0;
while (i < n) {
    System.out.println(computeElement(i));
    i++;
}

int i=0;
do {
    System.out.println(computeElement(i));
    i++;
} while (i<n);
```

 works since $n \geq 1$

Ornit Dror

22

for vs. while

- The following two statements are equivalent:

```
for(int i = 0 ; i < n ; i++)
    System.out.println(computeElement(i));

int i=0;
while (i < n) {
    System.out.println(computeElement(i));
    i++;
}
```

Ornit Dror

21