

## Αντικειμενοστρεφής Προγραμματισμός

**Παναγιώτης Αδαμίδης**  
adamidis@ihu.gr



**Wrapper classes, Τοποθέτηση δεδομένων και κλήση μεθόδων**

## Wrapper Classes

- Γενίκευση των 8 βασικών τύπων: Boolean, Byte, Character, Short, Integer, Long, Float, Double
- Καλούνται έτσι επειδή περικλείουν ένα βασικό τύπο έτσι ώστε μία μεταβλητή να μπορεί να αναπαρασταθεί ως αντικείμενο όταν είναι απαραίτητο.
- Επίσης παρέχουν την ελάχιστη και την μέγιστη τιμή (MAX\_VALUE, MIN\_VALUE) και για τους δύο τύπους κινητής υποδιαστολής (float, double) ορίζουν τις σταθερές POSITIVE\_INFINITY, NEGATIVE\_INFINITY και NaN.

## Wrapper Classes - Παράδειγμα

```
public class TestShort {
    public static void main(String[] args) {
        short m=22;
        System.out.println ("short m = " + m);
        Short x = new Short(m); // short --> Short
        System.out.println ("Short x = " + x);
        String s = x.toString(); // Short --> String
        System.out.println ("String s = " + s);
        m = Short.parseShort(s); // String --> short
        System.out.println ("short m = " + m);
        s = Short.toString(m); // short --> String
        System.out.println ("String s = " + s);
        x = Short.decode(s); // String --> Short
        System.out.println ("Short x = " + x);
        m = x.shortValue(); // Short --> short
        System.out.println ("short m = " + m);
        System.out.println ("Short.MIN_VALUE = " + Short.MIN_VALUE);
        System.out.println ("Short.MAX_VALUE = " + Short.MAX_VALUE);
    }
}
```

## Τοποθέτηση Δεδομένων

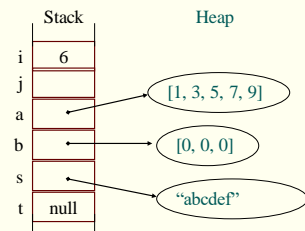
- Καταχωρητές (Registers): Μόνο ο μεταγλωττιστής έχει πρόσβαση.
- Stack: Χώρος στη RAM. Ο επεξεργαστής τον υποστηρίζει μέσω ενός δείκτη (stack pointer). Πολύ γρήγορη ανάθεση μνήμης. Ο μεταγλωττιστής πρέπει να γνωρίζει τον τύπο, το μέγεθος και την διάρκεια ζωής των δεδομένων που θα αποθηκευτούν. Αποθήκευση αναφορών σε αντικείμενα και βασικών τύπων δεδομένων.
- Heap: Γενικός χώρος μνήμης όπου «ζουν» όλα τα αντικείμενα της Java. Επίσης χώρος στη RAM. Πιο ευέλικτο μέρος από το Stack. Δεν χρειάζεται να γνωρίζουμε μέγεθος και διάρκεια ζωής των δεδομένων. Περισσότερος χρόνος για να γίνει ανάθεση μνήμης από ότι στο stack.
- Static storage: Χώρος στη RAM. Περιέχει δεδομένα τα οποία είναι διαθέσιμα καθ' όλη την διάρκεια του προγράμματος. Τα αντικείμενα δεν τοποθετούνται εδώ.

## Stack - Heap

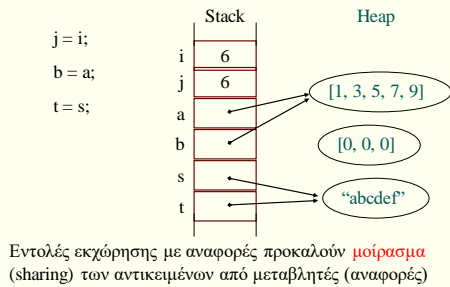
- **Stack:** Τοπικές μεταβλητές (ο χώρος τους δίνεται όταν καλείται η μέθοδος και αποδεσμεύεται με το τέλος της μεθόδου), Βασικοί τύποι (int, float κλπ.), αναφορές σε αντικείμενα
- **Heap:** Αντικείμενα (τελεστής new) - Garbage collector vs C++ memory leaks

## Παράδειγμα: Stack – Heap

```
int i=6;
int j;
int[] a = {1,3,5,7,9};
int[] b = new int[3];
String s = "abcdef"
String t = null;
```

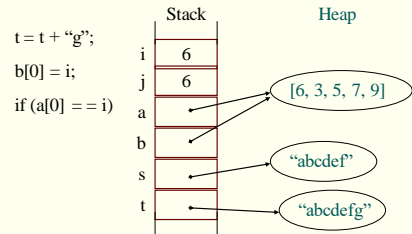


### Παράδειγμα: Stack – Heap (2)



### Δυνατότητα μεταβολής

➤ Άλλα αντικείμενα μπορούν να μεταβληθούν (mutable) και άλλα όχι (immutable)



### Κλήση μεθόδων: Σημαντική

- Κλήση μεθόδου a.m(...):
  - υπολογίζεται από το **a** η κλάση ή το αντικείμενο στο οποίο ανήκει η μέθοδος. Εάν η τιμή του a είναι null τότε δεν αναφέρεται σε κάποιο αντικείμενο (NullPointerException)
  - Από τα ορίσματα υπολογίζονται οι **πραγματικές** παράμετροι (actual parameters)
  - Δημιουργία **εγγραφής ενεργοποίησης** (activation record) στο stack. Η εγγραφή έχει χώρο για τις **τυπικές** παραμέτρους της μεθόδου και άλλες τοπικές μεταβλητές.
  - Οι τιμές των πραγματικών παραμέτρων δίνονται στις τυπικές (call by value)
  - Ο έλεγχος μεταφέρεται στην καλούμενη μέθοδο.

### Κλήση μεθόδων: Παράδειγμα

➤ Υποθέτουμε ότι η κλάση Arrays περιέχει την μέθοδο multiplies η οποία πολλαπλασιάζει κάθε στοιχείο ενός πίνακα με ένα αριθμό m.

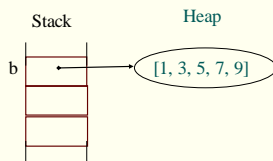
```

public static void multiplies (int[] a, int m) {
    if (a == null) return;
    for (int i=0; i<a.length; i++)
        a[i] = a[i]*m;
}
    
```

### Κλήση μεθόδων: Παράδειγμα (2)

➤ Παρακάτω δίνεται η κατάσταση πριν την κλήση και μετά την δημιουργία του πίνακα με την εντολή:

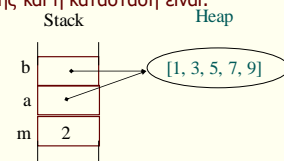
```
int [] b = {1, 3, 5, 7, 9}
```



### Κλήση μεθόδων: Παράδειγμα (3)

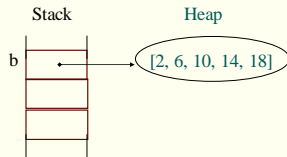
➤ Η μέθοδος καλείται ως εξής:  
 Arrays.multiplies(b, 2)

➤ Μετά την κλήση ο Stack περιέχει την εγγραφή ενεργοποίησης και η κατάσταση είναι:



### Κλήση μεθόδων: Παράδειγμα (4)

- Με το τέλος της μεθόδου η εγγραφή ενεργοποίησης έχει απορριφθεί.
- Το όρισμα πίνακας έχει μετατραπεί και αυτό είναι ορατό από την μεταβλητή b.
- Με το τέλος της εκτέλεσης της μεθόδου η κατάσταση είναι:



### Ερωτήσεις;

