



ΕΡΓΑΣΤΗΡΙΟ 5

(Memory Managment)

1. Επίδειξη εργαλείου VisualVM.

- Εγκατάσταση του plugin Visual GC. (Χρησιμοποιήστε το σύνδεσμο: <https://visualvm.github.io/uc/release139/updates.xml.gz>)
- Επεξήγηση οργάνωσης μνήμης.
- Profiler

Χρησιμοποιήστε το παρακάτω κώδικα για δοκιμές:

```
import java.util.Scanner;
public class DynamicStringConcatenation {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);

        String result = "";
        for (int i = 0; i < 100000; i++) {
            StringBuilder tmp = new StringBuilder();
            tmp.append(result);
            tmp.append("some more data");
            result = tmp.toString();
        }
        System.out.println(result);
        int n = scan.nextInt();
    }
}
```

2. Σας δίνεται το παρακάτω πρόγραμμα:

```
public class Memory {

    public static void main(String[] args) { // Line 1
        int i=1;                               // Line 2
        Object obj = new Object();             // Line 3
        Memory mem = new Memory();             // Line 4
        mem.foo(obj);                           // Line 5
    }                                           // Line 9

    private void foo(Object param) {           // Line 6
        String str = param.toString();         // Line 7
        System.out.println(str);               // Line 8
    }
}
```

Εξηγήστε μέσω ενός σχήματος, πως είναι οργανωμένη η μνήμη που το λειτουργικό σύστημα αναθέτει σε αυτό το πρόγραμμα.



3. Τι θα εκτυπώσει το παρακάτω πρόγραμμα; Γιατί;

```
public class StringPool {  
  
    /**  
     * Java String Pool example  
     * @param args  
     */  
    public static void main(String[] args) {  
        String s1 = "Cat";  
        String s2 = "Cat";  
        String s3 = new String("Cat");  
  
        System.out.println("s1 == s2 :"+(s1==s2));  
        System.out.println("s1 == s3 :"+(s1==s3));  
    }  
}
```

4. Τι θα εκτυπώσει το παρακάτω πρόγραμμα; Γιατί;

```
public static void main(String[] args) {  
    String x = new String("ab");  
    change(x);  
    System.out.println(x);  
}  
  
public static void change(String x) {  
    x = "cd";  
}
```

5. Το μέγεθος στη μνήμη ενός αντικειμένου εξαρτάται από την αρχιτεκτονική, κυρίως από το αν το VM είναι 32 ή 64 bit. Υπολογίστε τη χρήση μνήμης (σε bytes) των ακόλουθων αντικειμένων στη Java:
- ένα αντικείμενο Java που περιέχει δύο μεταβλητές int, μία μεταβλητή boolean, ένα αντικείμενο Long και μια αναφορά σε άλλα αντικείμενα.
 - Ένας πίνακας Java που αποτελείται από 20 αντικείμενα Integer.
6. (String tokenizer) Σχεδιάστε μια κλάση TokenizableString η οποία επιτρέπει την εξαγωγή και την εμφάνιση των λέξεων μιας πρότασης. Ένα αντικείμενο TokenizableString χαρακτηρίζεται από:
- Το περιεχόμενο (String);
 - ... ορίστε εσείς τα άλλα χαρακτηριστικά ενός τέτοιου αντικειμένου
 - Ορίστε τις λειτουργίες της κλάσης