

Προετοιμασία JAVA 2022

Ασκήσεις Αναδρομής - Μαθηματικών (1α και 2α Θέματα) (2-3hours)

- **Παραγοντικό** (factorial) - 2020K
 - for loop - αναδρομή
- **ΜΚΔ** (gcd) - 2017K - 2018K - 2020K
 - Αλγόριθμος του Ευκλείδη - Αναδρομή
- **ΕΚΠ** (lcm) - 2017E
 - $\text{lcm}(a, b) = a*b/\text{gcd}(a,b)$
- **Fibonacci** - 2017K - 2019K
 - Αναδρομή
- **Αποσταση**(vector, point)
- **Multiply**(vector, scalar) - 2020K
- **Multiply**(vector, vector) - 2020K
- **modulo** - 2018P
 - while and substractions
- **power of 2** - 2015E - 2017E - 2018K
 - for loop - αναδρομή
- **isCube** - 2014K
 - for loop
- **equation** - 2018P

Ασκήσεις με πίνακες (2α και 3α Θέματα) (2hours)

2d to 1d

- i-th element = (sum of i line === odd/even) - 2015E
- i-th element = max of i-th row/column - 2017K
- i-th element = sum of i-th row/column - 2018K

2d to 2d

- keep odd/even rows/columns - 2017E

1d to 1d

- add 2 1d arrays - 2018P
- cone array - 2020K

1d to scalar

- max/min of 1d array
- max/min of 1d array from i to j - 2018P - 2020K
- sum of elements
- sum of odd/even elements - 2019K

Άλλες

- print right triangle using * - 2014K

Ασκήσεις με Strings (3α και 4α Θέματα) (2hours)

- **isSubString** (η λέξη υπάρχει ακριβώς) - 2015E - 2018K
- **isWordInString** (η λέξη υπάρχει, αλλά επιτρέπονται και ενδιάμεσα γράμματα) - 2017K
- **bin2dec** - 2019K

Ασκήσεις Κλάσεων(4α και 5α θέματα) (2hours)

- **Point** και **Line/Circle** - 2019K - 2020K
- **Συνεδεμένη Λίστα** - 2017K - 2017E - 2018K - 2019K
- **MyResizableIntArrayList** - 2014K

Ασκήσεις με τυχαιότητα (4α και 5α θέματα) (20min)

- **Κληρωτιδα** (πχ joker) - 2017E
- **Ζάρι** (dice) - 2015E
- Άλλα παιχνίδια όπως ΠΜΨΧ

Άλλα θέματα (1hour)

- Θέμα 4ο Επαναληπτικής 2015
- **Πολλαπλής** - 2021K
 - Q16 (Understanding private members)
 - Q17 (understanding multiple constructors)
 - Q18 Creating new objects
 - Q19 silly trick
 - Q22 wrong for
 - Q23 double i++
 - Q25 create tringular matrix / run by hand

Χρήσιμα

- Δημιουργία μεταβλητών
 - int (το ίδιο και για float)

```
// 1ος τρόπος
int a; // πρώτη δημιουργώ μια κενή μεταβλητή
a = 14; // μετα αναθέτω μια τιμή (χωρίς το int)
// 2ος τρόπος
int a = 14; // τα κάνω όλα σε μια γραμμή
```

- char

```
char c = 'a'; // σωστο
char c = 'ab'; // ΛΑΘΟΣ - μονούς χαρακτήρες μόνο
char c = "a"; // ΛΑΘΟΣ - μονά αυτάκια
char c = "\u0008" // σωστό - 0 χαρακτήρας \u
```

ακολουθούμενος από ένα

// 'κουλό' νούμερο, μεταφράζεται σε έναν
μονό χαρακτήρα

- String ()

```
String str = "abc"; // σωστό
String str = 'abc'; // ΛΑΘΟΣ, εδώ θέλουμε διπλά αυτάκια
// Η κλάση String (την οποία κάποιος άλλος έχει
φτιαξει)
// περιέχει πολλές μεθόδους, κάποιες απ τις οποίες
είναι η
// length() και η charAt(int i)
int length = str.length() // length is 3
char firstLetter = str.charAt(0); // firstLetter is 'a'
char lastLetter = str.charAt(length-1); // secondLetter
is 'c'
```

- StringBuffer()

```
// Είναι ακριβώς ίδια με την String, με την μονη διαφορά
// οτι σου δίνει την δυνατότητα να τροποποιήσεις τα
γράμματα του
// string. Περιέχει την extra μέθοδο setCharAt(int i, char
c)
StringBuffer strB = "abc";
char firstLetter = str.charAt(0); // firstLetter is 'a'
strB.setCharAt(0, 'z');
char newFirstLetter = str.charAt(0); // newFirstLetter is
'z'
```

- Δημιουργία πινάκων

- Μονοδιάστατοι - 1D

```
// 1ος τρόπος
int array1d[]; // Δηλώνω οτι ο array1d θα είναι ένας
μονοδιάστατος πίνακας (διάνυσμα) που θα περιέχει integers
array1d = new int[10]; // Αναθέτω στον array1d 10 θέσεις
μνήμης
// Ο array1d έχει πλέον μέγεθος 10.

// 2ος τρόπος
int[] array1d;
array1d = new int[10]; //

// 3ος τρόπος
int[] array1d = new int[10]; // όλα σε μία γραμμή
```

```
// Αν ξέρεις απο πριν τι τιμές θες να βάλεις στον πίνακα
τότε
int[] array1d = new int[]{ 1,2,3,4,5,6,7,8,9,10 };

// Αντί για int, μπορούμε να έχουμε οποιονδήποτε άλλο τύπο
// είτε βασικούς τύπους της java
char[] array1d = new char[10];
float[] array1d = new float[10];
// είτε 'μη' βασικούς τύπους της java
String[] array1d = new String[10];
// είτε τύπους που δημιουργήσαμε εμείς
Person[] array1d = new Person[10];
// ο τύπος ενός πίνακα δηλώνει τι τύπου θα είναι τα
στοιχεία του πίνακα αυτού
```

- Δισδιάστατοι - 2D

```
// 3ος τρόπος (οι άλλοι δύο είναι και αυτοί όμοιοι με πριν)
int[][] array2d = new int[10][20]; // δισδιάστατος πίνακας
10 γραμμών και 20 στηλών
int[][] array2d = new int[10][]; // δισδιάστατος πίνακας 10
γραμμών - οι στήλες είναι άγνωστες ακόμα
// Ας φτιάξουμε έναν τριγωνικό πίνακα.
// Η 1η γραμμή έχει 1 στήλη,
// Η 2η γραμμή έχει 2 στήλες, κ.ο.κ
for (int i=0; i<10; i++){
    array2d[i] = new int[i+1]
    // array2d[i] είναι η i-οστή γραμμή του πίνακα array2d
    // new int[i+1] δημιουργεί έναν ΜΟΝΟδιάστατο πίνακα
    διαστάσεων i+1
    // Ένας ΔΥΣδιάστατος πίνακας μπορεί να θεωρηθεί σαν
    ένας ΜΟΝΟδιάστατος, όπου κάθε στοιχείο του είναι ένας άλλος
    ΜΟΝΟδιάστατος πίνακας.
}
```

- Κλάσεις

- Απλό παράδειγμα - Τα Βασικά

```
// 'Εστω ότι έχουμε φτιάξει την κλάση Bunny()
class Bunny{
    // πεδία (τα κάνω public, για να μην γράφω setters και
    getters)
    public int age;
    public String name;
    // Κατασκευαστή(ές)
    // Οι κατασκευαστές έχουν το ΙΔΙΟ όνομα με την κλάση.
    // Αν το όνομα δεν είναι ίδιο, ΔΕΝ ΕΙΝΑΙ ΚΑΤΑΣΚΕΥΑΣΤΕΣ
    public Bunny(int ageUser, String nameUser){

        // Αποθηκεύουμε την τιμή του χρηστή ageUser, στο
        πεδίο age της κλάσης. Η ageUser θα μπορούσε να λέγεται και
```

```

age, μην σε μπερδεψει
    // Εμείς Χρησιμοποιούμε το 'this' για να
αναφερθούμε στα πεδία της κλάσης (δεν είναι όμως
υποχρεωτικό)
    this.age = ageUser; // this == αυτηνης, ετουτης
    this.name = nameUser;
}
public void printInfo(){
    System.out.println(this.name, "is", this.age,
"year's old");
}
}

// Δημιούργησε το αντικείμενο heracles που είναι τύπου
Bunny. Το όνομά του είναι "Heracles" και η ηλικία του είναι
2 χρονών.
// Η λέξη 'new', δημιουργεί ένα αντικείμενο τύπου Bunny,
καλώντας τον κατάλληλο κατασκευαστή της Bunny.
Bunny heracles = new Bunny(2, "Heracles");

// Χρησιμοποιούμε την τελεία '.' για να προσπελάσουμε ένα
πεδίο ή μια μέθοδο ενός αντικειμένου.
print(heracles.name); // prints "Heracles"
print(heracles.age); // prints 2
heracles.printInfo(); // prints "Heracles is 2 year's old"
// εδώ δεν χρειάστηκε να πουμε print(heracles.printInfo()),
καθώς η μέθοδος printInfo α) έχει το print μέσα της και β)
δεν επιστρέφει τιποτα (void).
// προσεξε επισης οτι βαλαμε παρενθέσεις στην printInfo(),
αλλά όχι στα name και age. Οι παρενθέσεις "καλούν". Μια
μεθοδος/συναρτηση μπορεί να "καλεστεί", σε αντίθεση με τα
πεδία.

// ΛΑΘΗ ΠΟΥ ΚΑΝΕΙΣ ΠΑΝΤΑ
name.heracles; // πατατα
Bunny.name; // το πεδίο name ανηκει στο αντικείμενο
heracles, και όχι στην κλάση Bunny.
heracles = new Bunny(2, "Heracles") // λειπει το Bunny απ
την αρχη

```

- Random() - Μια άλλη κλάση από την βιβλιοθήκη java.util

```

// Πρεπει να την κάνουμε import
import java.util.Random;
// Δημιουργούμε μια γεννήτρια τυχαίων αριθμων.
// Ακριβώς με τον ίδιο τρόπο που δημιουργήσαμε πριν τον
Ηρακλή
Random generator = new Random();
// Ο λόγος που οι παρενθέσεις είναι κενές, είναι οτι ο
δημιουργός της Random, δεν χρειαζόταν κάποια είσοδο (όπως
πχ θέλαμε ονομα και ηλικία για την κλάση Bunny)

// Για να πάρουμε έναν τυχαίο αριθμοό απ το 0 ως το N, θα

```

```
χρησιμοποιήσουμε την μέθοδο nextInt(N)
int N = 10;
int newRandomNumber = generator.nextInt(N);
// Πάλι με την χρήση της τελείας αποκτάμε πρόσβαση στην
μέθοδο nextInt του generator, και την καλούμε
χρησιμοποιώντας τις παρενθέσεις () και το όρισμα εισόδου N.
Το αποτέλεσμα, το αποθηκεύουμε σε μιά καινούρια μεταβλητή
newRandomNumber.

// Για να πάρουμε τον επόμενο τυχαίο αριθμό,
newRandomNumber = generator.nextInt(N);
// Πρόσεξε ότι τώρα δεν έβαλα int πριν το newRandomNumber.
Το int (και γενικά ο τύπος) χρειάζεται μόνο την πρώτη φορά
που δημιουργείς μια μεταβλητή.
```