

Αλγοριθμική και Προγραμματισμός – Εντολές επανάληψης (WHILE & DO ... WHILE)

Άσκηση 1

Να γραφεί Πρόγραμμα σε Java, το οποίο:

1. **Διαβάζει** απ' το πληκτρολόγιο με την κλήση της μεθόδου `getInteger()`, η οποία βρίσκεται στην κλάση `ScannerUserInput`, μια τιμή σε μια ακέραια μεταβλητή `num`.
2. **Για όσο** η τιμή είναι μικρότερη του 2 και μεγαλύτερη του 5, το πρόγραμμα ζητάει νέα τιμή.
3. Το πρόγραμμα τερματίζει, όταν ο αριθμός που θα δημιουργηθεί είναι μεταξύ του 2 και του 5.
4. Τα βήματα 1,2,3 να υλοποιηθούν και με την εντολή **do...while**.
5. Μετά τη δημιουργία του σωστού αριθμού `num` στο $[2, 5]$, θα **υπολογίζει** και θα **εμφανίζει** όλες τις δυνάμεις του αριθμού $(num)^2, (num)^3, \dots$ για όσο η **Τιμή της Δύναμης** δεν έχει ξεπεράσει το **1000**, υπολογίζοντας την κάθε νέα **τιμή της δύναμης** από την **προηγούμενη τιμή επί τον αριθμό** και εμφανίζοντάς την.

✚ Το πρόγραμμα θα περιέχει και τα σχόλια που θα πρέπει να γράφονται σε κάθε Εργαστηριακή Άσκηση.

ΑΛΓΟΡΙΘΜΟΣ

Διαβάζω απ' το πληκτρολόγιο μια τιμή για τον αριθμό `num`

Για Όσο (ο αριθμός είναι εκτός των ορίων 2- 5)

Διαβάζω απ' το πληκτρολόγιο μια τιμή για τον αριθμό `num`

Κάνε τα παρακάτω

Διαβάζω απ' το πληκτρολόγιο μια τιμή για τον αριθμό `num`

Για Όσο (ο αριθμός είναι εκτός των ορίων 2- 5)

Για όσο (η Τιμή της Δύναμης δεν ξεπέρασε την τιμή 1000)

Υπολογισμός Επόμενης Τιμής της Δύναμης (Τιμή Δύναμης * Αριθμός)

Εμφάνιση της νέας Τιμής της Δύναμης

Άσκηση 2

Να τροποποιηθεί η **Άσκηση 1**, έτσι ώστε ο υπολογισμός των τιμών των δυνάμεων, όπως και στην **Άσκηση 1**, να γίνει με τη χρήση ενός εκθέτη, ο οποίος θα παίρνει της τιμές 1, 2, ... και τη Μέθοδο της Μαθηματικής Βιβλιοθήκης `Math.pow(<Βάση>, <Εκθέτης>)`. Η υλοποίηση να γίνει με την εντολή **while**. **Προσοχή**, η Μέθοδος `Math.pow()` επιστρέφει αποτέλεσμα τύπου `double`.

Άσκηση 3

Να τροποποιηθεί η **Άσκηση 2**, έτσι ώστε ο υπολογισμός των τιμών των δυνάμεων να γίνει με τη χρήση ενός **εκθέτη**, ο οποίος θα παίρνει της τιμές 1, 2, ... και τη Μέθοδο της Μαθηματικής Βιβλιοθήκης `Math.pow(<Βάση>, <Εκθέτης>)`. Η υλοποίηση να γίνει με την εντολή **do while**.

Ενδεικτική Έξοδος Προγράμματος

run:

```
Ο Αριθμός είναι : 1
Ο Αριθμός είναι : 9
Ο Αριθμός είναι : 8
Ο Αριθμός είναι : 10
Ο Αριθμός είναι : 5
```

```
Η Δύναμη είναι : 5
Η Δύναμη είναι : 25
Η Δύναμη είναι : 125
Η Δύναμη είναι : 625
```

```
while : Η 1-η Δύναμη με pow είναι : 5
while : Η 2-η Δύναμη με pow είναι : 25
while : Η 3-η Δύναμη με pow είναι : 125
while : Η 4-η Δύναμη με pow είναι : 625
```

```
do while : Η 1-η Δύναμη με pow είναι : 5
do while : Η 2-η Δύναμη με pow είναι : 25
do while : Η 3-η Δύναμη με pow είναι : 125
do while : Η 4-η Δύναμη με pow είναι : 625
do while : Η 5-η Δύναμη με pow είναι : 3125
```

BUILD SUCCESSFUL (total time: 0 seconds)

Άσκηση 4

Σε ένα διαγωνισμό του ΑΣΕΠ εξετάζονται 100 υποψήφιοι. Ως εξεταστικό κέντρο χρησιμοποιείται ένα κτίριο με αίθουσες διαφορετικής χωρητικότητας. Ο αριθμός των επιτηρητών που απαιτούνται ανά αίθουσα καθορίζεται αποκλειστικά με βάση τη χωρητικότητα της αίθουσας ως εξής:

| ΧΩΡΗΤΙΚΟΤΗΤΑ | ΑΡΙΘΜΟΣ ΕΠΙΤΗΡΗΤΩΝ |
|----------------------------|--------------------|
| Μέχρι και 15 θέσεις | 1 |
| Από 16 μέχρι και 23 θέσεις | 2 |
| Πάνω από 23 θέσεις | 3 |

Να γίνει πρόγραμμα σε Java το οποίο :

- Για κάθε αίθουσα, θα **διαβάζει** απ' το πληκτρολόγιο με την κλήση της μεθόδου `getInteger()`, η οποία βρίσκεται στην κλάση `ScannerUserInput`, μια τιμή σε μια ακέραια μεταβλητή `xoritikotita`.
- Θα **υπολογίζει** και θα **εμφανίζει** τον αριθμό των **επιτηρητών** που χρειάζονται για κάθε αίθουσα.
- Θα **σταματάει** όταν εξασφαλισθεί ο απαιτούμενος συνολικός αριθμός θέσεων. Να χρησιμοποιήσετε την εντολή `do...while`.
- Στο τέλος θα **εμφανίζει** το **σύνολο** των **αιθουσών**, των **επιτηρητών** και των **θέσεων** που χρησιμοποιήθηκαν.

Ενδεικτική Έξοδος Προγράμματος

run:

```
Δώσε χωρητικότητα αίθουσας : 15
Απαιτούνται 1 επιτηρητές
Δώσε χωρητικότητα αίθουσας : 20
Απαιτούνται 2 επιτηρητές
Δώσε χωρητικότητα αίθουσας : 25
Απαιτούνται 3 επιτηρητές
Δώσε χωρητικότητα αίθουσας : 23
Απαιτούνται 2 επιτηρητές
Δώσε χωρητικότητα αίθουσας : 16
Απαιτούνται 2 επιτηρητές
Δώσε χωρητικότητα αίθουσας : 12
Απαιτούνται 1 επιτηρητές
Σύνολο Αιθουσών : 6, Σύνολο Επιτηρητών : 11, Σύνολο Θέσεων : 111
BUILD SUCCESSFUL (total time: 0 seconds)
```

Οδηγίες κατάθεσης ασκήσεων

1. Συνδεθείτε στο URL: <http://aetos.it.teithe.gr/s>.
2. Επιλέξτε το μάθημα “Αλγοριθμική και Προγραμματισμός – Εργαστήριο X” (Όπου X ο αριθμός του εργαστηρίου του οποίου τις ασκήσεις πρόκειται να καταθέσετε) και πατήστε επόμενο
3. Συμπληρώστε τα στοιχεία σας. Πληκτρολογήστε USERNAME και PASSWORD ανάλογα με το τμήμα που παρακολουθείτε βάσει του παρακάτω πίνακα :

| | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 |
|----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| USERNAME | 00001 | 00002 | 00003 | 00004 | 00005 | 00006 | 00007 | 00008 | 00009 | 00010 |
| PASSWORD | 10000 | 20000 | 30000 | 40000 | 50000 | 60000 | 70000 | 80000 | 90000 | 10000 |

4. Αν θέλετε να καταθέσετε μόνο ένα java αρχείο μη το βάζετε σε zip file. Αντίθετα, αν θέλετε να καταθέσετε περισσότερα από ένα αρχεία, τοποθετήστε τα σε ένα zip ή rar file.
5. Αν χρησιμοποιείτε NetBeans, μη στέλνετε ολόκληρο το projects. Στείλτε μόνο .java αρχεία. Θα τα βρείτε στο φάκελο `src` στο φάκελο του project σας.
6. Επιλέξτε το αρχείο που θέλετε να στείλετε επιλέγοντας “choose file” στο πεδίο FILE1 και πατήστε “Παράδοση”.