DIDAKTIKH

02 ΚΑΤΑΝΟΗΣΗ

>_ Δρ. Στυλιανός Καραγιάννης, Ιόνιο Πανεπιστήμιο,
skaragiannis@ionio.gr

Βήμα Πρώτο: Δημιουργία Κινήτρου για Μάθηση
Στόχος: Ενεργοποίηση του ενδιαφέροντος των μαθητών.
Πώς να επιτευχθεί:

- ❑ Παρουσίαση ρεαλιστικών προβλημάτων.
- ❑ Σύνδεση με τις ανάγκες και τα ενδιαφέροντα των μαθητών.
- ❑ Χρήση παραδειγμάτων από την καθημερινή ζωή.

Βήμα Δεύτερο: Οικοδόμηση της Γνώσης μέσω Διερεύνησης και Συνεργασίας
Στόχος: Κατανόηση νέων εννοιών.
Πώς να επιτευχθεί:

- ❑ Ενεργή συμμετοχή των μαθητών σε διερευνητικές δραστηριότητες.
- ❑ Συνεργασία με άλλους για ανταλλαγή ιδεών και απόψεων.
- ❑ Υποστήριξη από τον εκπαιδευτή για την καθοδήγηση της διαδικασίας.

Βήμα Τρίτο: Εφαρμογή και Αναδόμηση της Γνώσης
Στόχος: Βαθύτερη κατανόηση και πρακτική εφαρμογή της γνώσης.
Πώς να επιτευχθεί:

- ❑ Εφαρμογή της γνώσης σε διαφορετικά πλαίσια.
- ❑ Αναστοχασμός για την αναδόμηση της γνώσης.
- ❑ Χρήση προβλημάτων επίλυσης και δημιουργικών δραστηριοτήτων.

Οι μαθητές καλούνται να κατανοήσουν και να συμπληρώσουν έναν μονοδιάστατο πίνακα ακολουθώντας μια αναδρομική σχέση, όπου κάθε στοιχείο του πίνακα (εκτός των δύο πρώτων) προκύπτει από το άθροισμα των δύο προηγουμένων.

Περιγραφή της διαδικασίας:

- **Αρχικοποίηση:** Ο πίνακας Α[10] έχει 10 στοιχεία, με τα δύο πρώτα να έχουν τιμή
- **Συμπλήρωση:** Κάθε επόμενο στοιχείο του πίνακα υπολογίζεται ως το άθροισμα των δύο προηγούμενων.

Στόχος της δραστηριότητας είναι να εξετάσουμε αν οι μαθητές είναι ικανοί να συμπληρώνουν τις τιμές ενός μονοδιάστατου πίνακα με βάση την περιγραφή της αναδρομικής σχέσης των στοιχείων του καθώς και να σχεδιάζουν αλγοριθμική επίλυση του προβλήματος της δημιουργίας αυτού του πίνακα.

Δίνεται ένας αταξινόμητος μονοδιάστατος πίνακας. Ο στόχος είναι να υπολογιστούν το μέγιστο και το ελάχιστο στοιχείο του πίνακα, χωρίς να γίνει πρώτα ταξινόμηση του πίνακα.

4	7	10	9	7	20	2	14	17	12

Το πλήθος των στοιχείων που απαιτείται αλγοριθμικά να προσπελάσουμε για να βρούμε το μέγιστο και το ελάχιστο στοιχείο του πίνακα είναι ίσο με:

- ΠΛΗΘΟΣ ΠΡΟΣΠΕΛΑΣΕΩΝ ΓΙΑ ΤΟ ΜΕΓΙΣΤΟ = ...
- ΠΛΗΘΟΣ ΠΡΟΣΠΕΛΑΣΕΩΝ ΓΙΑ ΤΟ ΕΛΑΧΙΣΤΟ = ...

Γιατί: .....

Σκοπός της Δραστηριότητας:
Να εξετάσουμε αν οι μαθητές μπορούν να προσδιορίσουν το πλήθος των στοιχείων που πρέπει να προσπελάσουμε αλγοριθμικά για να βρούμε το μέγιστο και το ελάχιστο ενός αταξινόμητου μονοδιάστατου πίνακα, χωρίς την ανάγκη ταξινόμησης.

Κατευθυντήριες Οδηγίες:

- Ο πίνακας περιέχει **N** στοιχεία.
- Οι μαθητές πρέπει να κατανοήσουν ότι για να βρούμε τόσο το μέγιστο όσο και το ελάχιστο στοιχείο, χρειάζεται να προσπελάσουμε όλα τα στοιχεία του πίνακα τουλάχιστον μία φορά.

Για να υπολογίσουμε το μέγιστο και το ελάχιστο στοιχείο σε έναν αταξινόμητο μονοδιάστατο πίνακα, χωρίς να τον ταξινομήσουμε, θα πρέπει να διασχίσουμε όλα τα στοιχεία του πίνακα.

- Για να βρούμε το μέγιστο στοιχείο, πρέπει να ελέγξουμε κάθε στοιχείο του πίνακα και να το συγκρίνουμε με το τρέχον μέγιστο.
- Ομοίως, για να βρούμε το ελάχιστο στοιχείο, πρέπει να συγκρίνουμε κάθε στοιχείο με το τρέχον ελάχιστο.

Υπολογισμός:

Έστω ότι ο πίνακας έχει **N** στοιχεία.

- **Πλήθος προσπελάσεων για το μέγιστο:** Για να βρούμε το μέγιστο, πρέπει να προσπελάσουμε **όλα τα στοιχεία** του πίνακα τουλάχιστον μία φορά, επομένως ο αριθμός των προσπελάσεων είναι **N**.
- **Πλήθος προσπελάσεων για το ελάχιστο:** Ομοίως, για να βρούμε το ελάχιστο, πρέπει να προσπελάσουμε **όλα τα στοιχεία** μία φορά, άρα και εδώ ο αριθμός των προσπελάσεων είναι **N**.

Γιατί:

- Σε έναν αταξινόμητο πίνακα δεν έχουμε καμία πληροφορία για τη διάταξη των στοιχείων, επομένως δεν μπορούμε να βρούμε το μέγιστο ή το ελάχιστο χωρίς να συγκρίνουμε κάθε στοιχείο.
- Για κάθε σύγκριση, πρέπει να κάνουμε μία προσπέλαση, και έτσι χρειαζόμαστε **N** προσπελάσεις τόσο για το μέγιστο όσο και για το ελάχιστο. Αντίθετα στον ταξινομημένο παρακάτω:

2	4	7	7	9	10	12	14	17	20

Ταξινόμηση φυσαλίδας

Θεωρείστε τον παρακάτω μονοδιάστατο πίνακα Α[5].

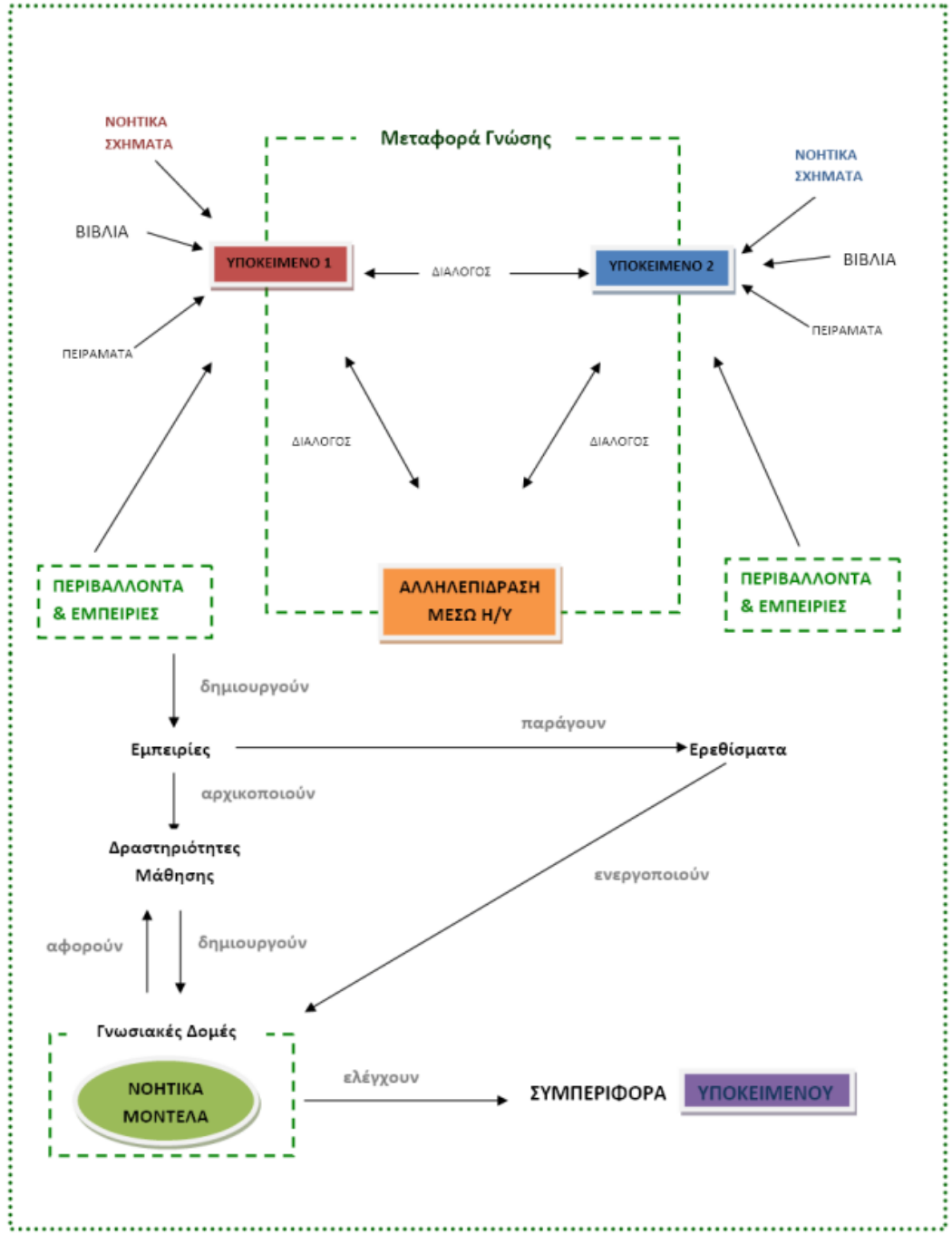
Να συμπληρώσετε τα δύο πρώτα περάσματα κατά τη διαδικασία ταξινόμησης του πίνακα Α με τη μέθοδο της φυσαλίδας.

Στόχος της δραστηριότητας είναι να εξετάσουμε αν οι μαθητές είναι ικανοί να εφαρμόζουν τον αλγόριθμο ταξινόμησης της φυσαλίδας.

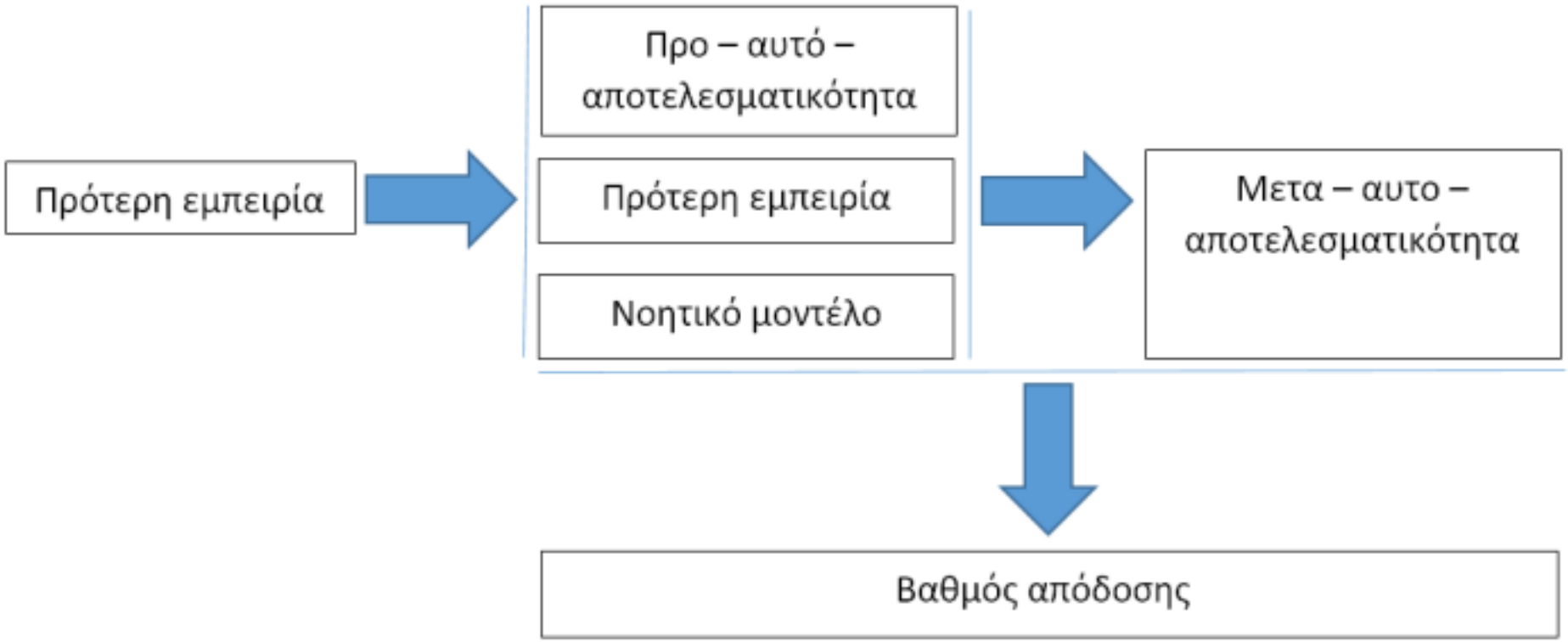
Αρχικός πίνακας: A = [4, 2, 5, 1, 3]

1ο πέρασμα:
Στο πρώτο πέρασμα, συγκρίνουμε συνεχόμενα στοιχεία και τα ανταλλάσσουμε όπου χρειάζεται:
1.Συγκρίνουμε **A[0]** (4) και **A[1]** (2). Επειδή 4 > 2, τα ανταλλάσσουμε.
Ο πίνακας γίνεται: [2, 4, 5, 1, 3]
2.Συγκρίνουμε **A[1]** (4) και **A[2]** (5). Επειδή 4 < 5, δεν κάνουμε καμία αλλαγή.
Ο πίνακας παραμένει: [2, 4, 5, 1, 3]
3.Συγκρίνουμε **A[2]** (5) και **A[3]** (1). Επειδή 5 > 1, τα ανταλλάσσουμε.
Ο πίνακας γίνεται: [2, 4, 1, 5, 3]
4.Συγκρίνουμε **A[3]** (5) και **A[4]** (3). Επειδή 5 > 3, τα ανταλλάσσουμε.
Ο πίνακας γίνεται: [2, 4, 1, 3, 5]

2ο πέρασμα:
Στο δεύτερο πέρασμα, επαναλαμβάνουμε την ίδια διαδικασία:
1.Συγκρίνουμε **A[0]** (2) και **A[1]** (4). Επειδή 2 < 4, δεν κάνουμε καμία αλλαγή.
Ο πίνακας παραμένει: [2, 4, 1, 3, 5]
2.Συγκρίνουμε **A[1]** (4) και **A[2]** (1). Επειδή 4 > 1, τα ανταλλάσσουμε.
Ο πίνακας γίνεται: [2, 1, 4, 3, 5]
3.Συγκρίνουμε **A[2]** (4) και **A[3]** (3). Επειδή 4 > 3, τα ανταλλάσσουμε.
Ο πίνακας γίνεται: [2, 1, 3, 4, 5]
4.Συγκρίνουμε **A[3]** (4) και **A[4]** (5). Επειδή 4 < 5, δεν κάνουμε καμία αλλαγή.
Ο πίνακας παραμένει: [2, 1, 3, 4, 5]



Εικόνα 3.1 *Η μεταφορά γνώσης (βασισμένο στο Μάρκελλος et al., 2012)*



Εικόνα 3.2 *Προτεινόμενο μοντέλο για τους παράγοντες που επηρεάζουν την απόδοση των εκπαιδευόμενων σε εισαγωγικά μαθήματα προγραμματισμού (βασισμένο στο Ramalingam, LaBelle & Wiedenbeck, 2004)*

Αδυναμία κατανόησης των εντολών ροής:

- **Λάθος**: Ο μαθητής πιστεύει ότι οι βρόχοι εκτελούνται μόνο μία φορά.
- **Παράδειγμα**:

```python
for i in range(5):
    print(i)  # Περιμένει ότι θα εκτυπωθεί μόνο το 0
```

- **Σωστό**: Ο βρόχος εκτελείται 5 φορές, εκτυπώνοντας τους αριθμούς από 0 έως 4.

Λανθασμένη αντίληψη των μεταβλητών:

- **Λάθος**: Ο μαθητής νομίζει ότι μια μεταβλητή μπορεί να αποθηκεύει πολλές τιμές ταυτόχρονα.
- **Παράδειγμα**:

```python
x = 5
x = 10  # Πιστεύει ότι το x κρατάει και τα δύο νούμερα
```

- **Σωστό**: Το x αποθηκεύει μόνο το 10 μετά την εκχώρηση.

2. Λανθασμένη αντίληψη των μεταβλητών

Άσκηση: Δημιούργησε έναν κώδικα που θα αποθηκεύει δύο διαφορετικές τιμές σε μία μεταβλητή `x`. Έλεγξε την τιμή του `x` μετά από κάθε εκχώρηση.

Οδηγίες:

```python
x = 5
print(x)  # Έλεγξε την τιμή εδώ
x = 10
print(x)  # Έλεγξε την τιμή εδώ
```

Λύση: Το `x` θα εκτυπώσει 5 στην πρώτη γραμμή και 10 στη δεύτερη.

Παρανόηση των αντικειμενοστραφών εννοιών:

- **Λάθος**: Ο μαθητής νομίζει ότι οι κλάσεις και τα αντικείμενα είναι το ίδιο πράγμα.
- **Παράδειγμα**:

```python
class Dog:
    def bark(self):
        return "Woof!"

my_dog = Dog()  # Πιστεύει ότι η Dog είναι το ίδιο με το my_dog
```

- **Σωστό**: Dog είναι η κλάση, ενώ my_dog είναι ένα αντικείμενο της κλάσης Dog.

Μη κατανόηση της αναδρομής:

- **Λάθος**: Ο μαθητής δεν καταλαβαίνει πώς μια συνάρτηση καλεί τον εαυτό της.
- **Παράδειγμα**:

```python
def factorial(n):
    if n == 1:
        return 1
    else:
        return n * factorial(n - 1)

print(factorial(5))  # Πιστεύει ότι το factorial θα εκτελείται μό
```

- **Σωστό**: Η συνάρτηση καλείται πολλές φορές, μέχρι να φτάσει στη βάση της αναδρομής.

6. Λανθασμένη κατανόηση του εύρους (scope) των μεταβλητών

Άσκηση: Γράψε μια συνάρτηση που θα δηλώνει μια τοπική μεταβλητή και μετά προσπάθησε να την εκτυπώσεις έξω από τη συνάρτηση.

Οδηγίες:

python Copy code

```python
def my_function():
    b = 10

my_function()
print(b)  # Τι θα συμβεί εδώ;
```

Λύση: Θα προκαλέσει σφάλμα επειδή η μεταβλητή `b` δεν είναι προσβάσιμη έξω από τη συνάρτηση.

7. Παρανόηση των αντικειμενοστραφών εννοιών

Άσκηση: Δημιούργησε μια κλάση `Car` και προσπάθησε να εκτυπώσεις τα χαρακτηριστικά της χωρίς να δημιουργήσεις αντικείμενο.

Οδηγίες:

python Copy code

```python
class Car:
    def __init__(self, model):
        self.model = model

print(Car.model)  # Πώς θα προσπαθήσεις να το εκτυπώσεις;
```

Λύση: Θα προκαλέσει σφάλμα. Πρέπει πρώτα να δημιουργήσεις ένα αντικείμενο της κλάσης:

python Copy code

```python
my_car = Car("Toyota")
print(my_car.model)
```

6. Λανθασμένη κατανόηση του εύρους (scope) των μεταβλητών

Άσκηση: Δημιούργησε μια συνάρτηση που θα προσπαθεί να αλλάξει μια καθολική μεταβλητή και δες αν αυτό λειτουργεί.

Οδηγίες:

python Copy code

```python
a = 5

def modify_a():
    a = 10  # Αυτή είναι τοπική ή καθολική;

modify_a()
print(a)  # Τι θα εκτυπωθεί εδώ;
```

Λύση: Θα εκτυπωθεί 5, επειδή η `a` μέσα στη συνάρτηση είναι τοπική.

1.Πρόβλημα: Ορισμός και ανάλυση προβλημάτων που μπορούν να λυθούν μέσω προγραμματισμού.
2.Μεταβλητές: Ορισμός και χρήση μεταβλητών για την αποθήκευση δεδομένων. Περιλαμβάνει τη δήλωση, τον τύπο και την αρχικοποίηση.
3.Πράξεις, Τελεστές: Βασικές αριθμητικές και λογιστικές πράξεις, όπως πρόσθεση, αφαίρεση, πολλαπλασιασμός και διαίρεση. Περιλαμβάνει και λογικούς τελεστές (AND, OR, NOT).
4.Απλά προβλήματα: Λύση απλών προβλημάτων μέσω προγραμματισμού, με παραδείγματα που εφαρμόζουν τις αρχές που έχουν διδαχθεί.
5.Δομή επιλογής: Χρήση δομών ελέγχου (if-else, switch) για να παίρνονται αποφάσεις στον κώδικα με βάση συνθήκες.
6.Δομή Επανάληψης: Χρήση βρόχων (for, while) για επαναλαμβανόμενες διαδικασίες και την εκτέλεση κώδικα πολλαπλές φορές.
7.Πίνακες: Δημιουργία και χρήση μονοδιάστατων πινάκων για την αποθήκευση και οργάνωση δεδομένων.
8.Δισδιάστατοι πίνακες: Οργάνωση δεδομένων σε μορφή πινάκων με δύο διαστάσεις, χρήσιμοι για εφαρμογές όπως οι πίνακες.

9. **Αναζήτηση:** Διαδικασίες αναζήτησης δεδομένων σε πίνακες, π.χ. γραμμική και δυαδική αναζήτηση.
10. **Ταξινόμηση:** Διαδικασίες ταξινόμησης δεδομένων, π.χ. μεθόδους όπως η φυσαλίδα (bubble sort) και η επιλογής (selection sort).
11. **Συναρτήσεις:** Δημιουργία και χρήση συναρτήσεων για την οργάνωση του κώδικα και την επαναχρησιμοποίηση κώδικα.
12. **Δομές Δεδομένων:** Εισαγωγή σε βασικές δομές δεδομένων όπως λίστες, στοίβες και δέντρα, με παραδείγματα χρήσης τους.
13. **Αντικειμενοστραφής Προγραμματισμός:** Βασικές αρχές του OOP (Object-Oriented Programming), όπως κλάσεις, αντικείμενα, κληρονομικότητα και πολυμορφισμός.

Επιπλέον Θέματα:

α. **Μετρητές:** Χρήση μετρητών σε βρόχους και για τον υπολογισμό των επαναλήψεων.
β. **ΜΟ:** Ορισμός και χρήση μέσων όρων και πώς σχετίζονται με τη δομή δεδομένων.
γ. **Κλιμακωτή χρέωση vs Κανονική:** Συγκριτική ανάλυση μεταξύ κλιμακωτής χρέωσης και σταθερής χρέωσης, π.χ. για υπηρεσίες ή προϊόντα.