Οι τύποι στη Java χωρίζονται σε δύο κατηγορίες:
• τους πρωταρχικούς τύπους (primitive types) και
• τους τύπους αναφοράς (reference types)

Οι **πρωταρχικοι τυποι** ονομαζονται και αλλιως βασικοι και χωριζονται στους αριθμητικους τυπους και στον λογικο τελεστη boolean.Οι αριθμητικοι τυποι με την σειρα τους χωριζονται στους τυπους διαστηματος (int) και στους τυπους κινητης υποδιαστολης (double) .

Οι **τυποι αναφορας** ανηκουν σε τρεις κατηγοριες:
- τυπος κλασης
- τυπος πινακα
- τυπος διασυνδεσης

Η κλαση αποτελει το βασικο μηχανισμο παραγωγης νεων τυπων δεδομενων απο το χρηστη.
Ο τυπος πινακα οριζεται με τη βοηθεια ενος ειδικου μηχανισμου που οριζεται απο την γλωσσα.
Ο τυπος διασυνδεσης αποτελει ενα μηχανισμο ορισμου αφηρημενων τυπων δεδομενων.

Χωρος αποθηκευσης δεδομενων στην JAVA:
- Στοιβα
- Σωρος
- Στατικη περιοχη μνημης
- Περιοχη σταθερων
- Καταχωρητες
- Δευτερευουσα Μνημη

Γραμμικη Δομη Δεδομενων
Ονομαζεται η Δ.Δ το συνολο των στοιχειων της οποιας ειναι διατεταγμενο με τετοιο τροπο ωστε να ισχυουν τα εξης: α) υπαρχει ενα στοιχειο το οποιο ονομαζεται αρχη και εχει ενα και μονον ενα επομενο στοιχειο , β) υπαρχει ενα στοιχειο το οποιο ονομαζεται τελος και εχει ενα και μονον ενα προηγουμενο στοιχειο , γ) καθε αλλο στοιχειο εχει ενα και μονον ενα προηγουμενο και ενα και μονον ενα επομενο.

Μη γραμμικη Δομη Δεδομενων
Πρόκειται για μια Δ.Δ καθε στοιχειο της οποιας μπορει να εχει πολλα επομενα στοιχεια.Σε μια μη γραμμικη Δ.Δ υπαρχει μια ιεραρχικη σχεση μεταξυ των δεδομενων, καθως υπαρχουν ριζες,παιδια και κομβοι.

Πραξεις/Λειτουργιες στις Γραμμικες Δ.Δ
ΒΑΣΙΚΕΣ:
- **Προσπελαση**: Προσβαση σε ενα στοιχειο της δομης με σκοπο να εξετασθει ή να τροποποιηθει το περιεχομενο του.
- **Εισαγωγη**: Προσθηκη νεου στοιχειου σε μια υπαρχουσα δομη.
- **Διαγραφη**: Η αφαιρεση ενος στοιχειου απο μια υπαρχουσα δομη.

ΕΠΙΠΛΕΟΝ:
- **Αναζητηση**: Πραγματοποιειται διαδοχικα προσπελαση στα στοιχεια μιας δομης, με στοχο να εντοπισθουν ενα ή περισσοτερα που εχουν μια ζητουμενη ιδιοτητα.
- **Ταξινομηση**: Τα στοιχεια μιας δομης αναδιατασσονται ωστε να τοποθετουνται σε αυξουσα ή φθινουσα σειρα.

Το μεγεθος ενος πινακα παραμενει σταθερο κατα την διαρκεια εκτελεσης του προγραμματος (στατικο μεγεθος) .

Γραμμικες Δ.Δ: πινακας , διανυσμα , συμβολοσειρα , στοιβα , ουρα, λιστα , συνδεδεμενη λιστα.
Μη γραμμικες ΔΔ: δεντρα,σωρος

Διανυσμα(Vector) : δεν ειναι στατικος ως προς το μεγεθος οπως ο πινακας,μπορει να μεταβαλει το μηκος του κατα την διαρκεια εκτελεσης τους προγραμματος.

Στοιβα (Stack): ειναι μια λιστα στην οποια νεα στοιχεια μπορουν να προστεθουν και να αφαιρεθουν μονο απο τη μια ακρης της (κορυφη στοιβας) . Ειναι τυπου LIFO (Last In First Out).

```
public interface Stack {
public int size( );                    // επιστρέφει το μέγεθος της στοίβας
public boolean isEmpty( );       // αληθεύει εάν η στοίβα είναι κενή
public Object top( ) throws StackEmptyException;
// επιστρέφει το στοιχείο που βρίσκεται στην κορυφή της στοίβας
public void push(Object item) throws StackFullException;
// εισάγει ένα νέο στοιχείο στην κορυφή της στοίβας
public Object pop( ) throws StackEmptyException;
// εξάγει και επιστρέφει το στοιχείο που βρίσκεται στην κορυφή της στοίβας
}
```

Εισαγωγή στοιχείου στη Στοίβα→ Κατάσταση εξαίρεσης (υπερχείλιση στοίβας)

```
public void push(Object item) throws StackFullException
{
if (size( )== capacity-1)
throw new StackFullException("Stack overflow");
S[++top] = item;
}
```

Εξαγωγή στοιχείου από Στοίβα→ κατάσταση εξαίρεσης (άδεια στοίβα)

```
public Object pop( ) throws StackEmptyException {
  Object element;
  if (isEmpty())
          throw new StackEmptyException("Stack is empty");
  element = S[top];
  S[top--] = null;
  return element;
}
```

Ουρα (queue) : ειναι μια λιστα στην οποια μπορουν να προστεθουν στοιχεια μονο στη μια ακρη (πισω) και να αφαιρεθουν απο την αλλη (μπροστα). Ειναι τυπου FIFO (First In First Out).

```
public interface Queue {
public int size( );                    // επιστρέφει το μέγεθος (αριθμός στοιχείων) της ουράς
public boolean isEmpty( );       // αληθεύει εάν η ουρά είναι κενή
public Object front( ) throws QueueEmptyException;
// επιστρέφει το στοιχείο που βρίσκεται στο εμπρός μέρος της ουράς
public void enqueue(Object item) throws QueueFullException;
// εισάγει ένα νέο στοιχείο στο πίσω μέρος της ουράς
public Object dequeue( ) throws QueueEmptyException;
// εξάγει και επιστρέφει το στοιχείο που βρίσκεται
// στο εμπρός μέρος της ουράς
```

Εισαγωγή στοιχείου στην Ουρά → κατάσταση εξαίρεσης (υπερχείλιση ουράς)

```
public void enqueue(Object item) throws QueueFullException  {
        if (last == capacity) /* Εικονική Υπερχείλιση?? */
                throw new QueueFullException("Queue overflow");
        Q[last++] = item;
}
```

Εξαγωγή στοιχείου από Ουρά→ κατάσταση εξαίρεσης (άδεια ουρά).

```
public Object dequeue( ) throws QueueEmptyException {
  Object item;
  if (isEmpty( ))
        throw new QueueEmptyException("Queue is empty");
  item = Q[first];
  Q[first++] = null; //!!! για τον garbage collector
  return item;
}
```

Μια δομη δεδομενων η οποια μπορει να δημιουργηθει και να μετατρεπεται στη διαρκεια της εκτελεσης του προγραμματος αυξομειωνοντας το μεγεθος της ή το χωρο που καταλαμβανει στη μνημη ονομαζεται **Δυναμικη Δομη Δεδομενων** (ουρα,στιβα,συνδεδεμενη λιστα,δεντρα).

Λιστα (list) ειναι ενα διατεταγμενο συνολο απο 0 ή περισσοτερα στοιχεια τα οποια κατα κανονα ειναι ολα του ιδιου τυπου. Διαφερει απο τον πινακα γιατι το μεγεθος της ειναι μεταβλητο.Ο αριθμός των στοιχείων της λίστας ονομάζεται μήκος (length) της λίστας.

Αναδρομικος ορισμος ΔΕΝΤΡΟ:
Δέντρο είναι ένα σύνολο Τ από κόμβους , τέτοιο ώστε είτε:
(α) Το Τ είναι κενό ή
(β) Το Τ περιλαμβάνει ένα ξεχωριστό κόμβο, R, που ονομάζεται ρίζα του Τ και οι υπόλοιποι κόμβοι Τ - {R} χωρίζονται σε μηδέν ή περισσότερα σύνολα κόμβων, Τ1, Τ2, . . ., Τn, που είναι ξένα μεταξύ τους και τα οποία είναι με τη σειρά τους δέντρα. Τα Τ1, Τ2, . . ., Τn, ονομάζονται υποδέντρα του Τ.

Δεντρο ειναι μια συλλογη απο στοιχεια, που ονομαζονται κομβοι.Οι κομβοι του δεντρου συνδεονται μεταξυ τους με την βοηθεια ακμων με βαση τους εξης κανονες: α) υπαρχει ενας και μονον ενας κομβος στον οποιο δεν καταληγει καμια ακμη (ριζα) , β) σε ολους τους υπολοιπους κομβους καταληγει υποχρεωτικα μια και μονο μια ακμη.

Καθε κομβος (εκτος της ριζας) εχει ακριβως εναν κομβο απο πανω του, ο οποιος ονομαζεται **πατερας**. Οι κομβοι που βρισκονται ακριβως κατω απο εναν κομβο, ονομαζονται **παιδια** του.

Ενας κομβος που δεν εχει κανενα μη-κενο υποδεντρο(κομβο χωρις παιδια) ονομαζεται **τερματικος κομβος ή φυλλο του δεντρου**. Ολοι οι υπολοιποι κομβοι ονομαζονται **μη τερματικοι**. Οι τερματικοι αναφερονται και σαν **εξωτερικοι** ενω οι μη τερματικοι σαν **εσωτερικοι.**
Μία ακολουθία κόμβων, ενός δέντρου, οι οποίοι συνδέονται διαδοχικά μεταξύ τους με τη βοήθεια ακμών, ονομάζεται μονοπάτι (path).
Οι κόμβοι ενός δέντρου χωρίζονται σε **επίπεδα** (levels). Ενας κόμβος βρίσκεται στο επίπεδο Ν, εάν Ν είναι ο αριθμός των ακμών του μονοπατιού, που συνδέει τη ρίζα με τον κόμβο αυτό.
Ορίζουμε σαν **ύψος** (heigth) ή **βάθος** (depth) ενός κόμβου τον αριθμό του επιπέδου στο οποίο βρίσκεται ο κόμβος αυτός. Το **ύψος** (ή βάθος) ενός δέντρου ταυτίζεται με το μέγιστο ύψος (ή βάθος) των κόμβων του δέντρου.

Ονομάζουμε **βαθμό** (degree) ενός κόμβου τον αριθμό των υποδέντρων του.
Ονομάζουμε **δάσος** (forest) ένα σύνολο από N >= 0 δέντρα που είναι ξένα μεταξύ τους.

Δυαδικό δέντρο (binary tree) είναι ένα δέντρο του οποίου κάθε κόμβος έχει το πολύ δύο υποδέντρα. Τα υποδέντρα του δυαδικού δέντρου ονομάζονται αριστερό και δεξιό υποδέντρο αντίστοιχα.
Βασικες Μεθοδοι διελευσης Δυαδικου Δεντρου:
1ος τροπος: Ενθεματικη Διελευση
Βημα 1:Αριστερο υποδεντρο
Βημα 2:Ριζα
Βημα 3:Δεξιο υποδεντρο

2ος τροπος: Προθεματικη διελευση
Βημα 1:Ριζα
Βημα 2:Αριστερο υποδεντρο
Βημα 3:Δεξιο υποδεντρο

3ος τροπος: Επιθεματικη διελευση
Βημα 1:Αριστερο υποδεντρο
Βημα 2:Δεξιο υποδεντρο
Βημα 3:Ριζα

Ένα δυαδικό δέντρο βάθους N ονομάζεται **πλήρες** (complete) όταν έχει όλους τους κόμβους του επιπέδου N συμπληρωμένους.

Ένα δυαδικό δέντρο βάθους N ονομάζεται **σχεδόν πλήρες** (almost complete) όταν έχει όλους τους κόμβους του επιπέδου N-1 συμπληρωμένους και οι κόμβοι που υπάρχουν στο N-οστό επίπεδο είναι τοποθετημένοι όσο το δυνατόν πιο αριστερά.

Ένα δυαδικό δέντρο ονομάζεται **σωρός** (heap) όταν ισχύουν:
(α)το δυαδικό δέντρο είναι σχεδόν πλήρες και
(β)οι τιμές των κόμβων είναι τοποθετημένες με τέτοιο τρόπο ώστε ο κάθε κόμβος πατέρας να έχει μεγαλύτερη τιμή από τα παιδιά του.

Ένα **αρχείο** (file) είναι μία σύνθετη δομή δεδομένων που αποτελείται από μία σειρά στοιχείων (elements), τα οποία είναι (συνήθως) του ίδιου τύπου.

Τα στοιχεία ενός αρχείου ονομάζονται και **λογικές εγγραφές ή απλά εγγραφές** (structures).

Αναλογα με τον τροπο αποθηκευσης:
α) αρχεια κειμενου , β) δυαδικα αρχεια

Οι πιο βασικοί τύποι αρχείων που μπορούν να οριστούν είναι τα **σειριακά ή ακολουθιακά αρχεία** (sequential flies) και τα **αρχεία κατ' ευθείαν πρόσβασης** (direct access).

Ρεύμα (Stream) είναι ένα αντικείμενο το οποίο αναπαριστά μία σειριακή ροή δεδομένων από μία πηγή προς έναν προορισμό.

Υπαρχουν δυο ειδη ρευματων:

α)**InOutStream**: χρησιμοποιεται για την αναγνωση δεδομενων απο μια πηγη
β)**OutPutStream**: χρησιμοποιειται για την εγγραφη δεδομενων σε καποιο προορισμο.

Ένα **ρεύμα φίλτρο** φιλτράρει δεδομένα όπως αυτά διαβάζονται ή γράφονται στο ρεύμα.

Ενα **ακολουθιακό αρχείο** μοιάζει, ως προς τη δομή του, με έναν πίνακα, αποτελείται δηλαδή από στοιχεία που είναι όλα του ίδιου τύπου, διαφέρει όμως από τον πίνακα σε δύο βασικά σημεία:
(α) το μέγεθος του , β) τον τρόπο πρόσβασης

Τα **ακολουθιακά αρχεία** κατά τη διάρκεια της επεξεργασίας τους ανοίγουν κατ' αποκλειστικότητα είτε σαν αρχεία εισόδου είτε σαν αρχεία εξόδου.

Ο όρος **αρχείο κατευθείαν πρόσβασης** (dirrect access file), χαρακτηρίζει ένα αρχείο, στο οποίο έχουμε τη δυνατότητα άμεσης πρόσβασης σ' ένα οποιδήποτε στοιχείο του , ονομάζεται και αρχείο τυχαίας πρόσβασης (random access file).