

Δομές Δεδομένων & Ανάλυση Αλγορίθμων 3ο Εξάμηνο

- **Αρχεία Κατ' ευθείαν Πρόσβασης**
(Direct Access Files) ή
(Random Access Files)

Δημοσθένης Σταμάτης

<http://www.iee.ihu.gr/~demos>

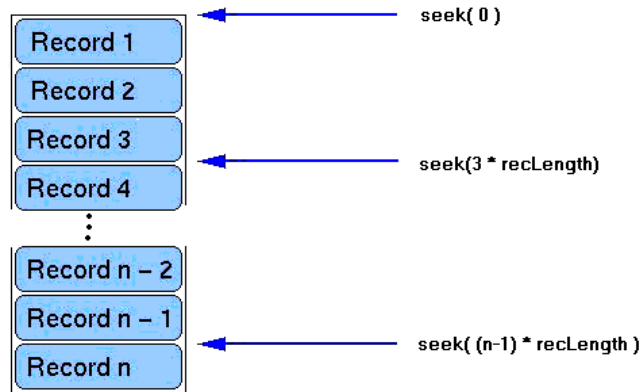
Τμήμα Μηχανικών Πληροφορικής & Ηλεκτρονικών Συστημάτων



Αρχεία Κατ' ευθείαν πρόσβασης Προϋποθέσεις που απαιτούνται:

1. Να είναι γνωστός ο τρόπος αρίθμησης των στοιχείων του αρχείου. Συνήθως η σχετική Θέση του πρώτου στοιχείου είναι 0 (μηδέν), ενώ η σχετική Θέση του τελευταίου στοιχείου ενός αρχείου που περιέχει N στοιχεία είναι $N-1$.
2. Να υπάρχει η δυνατότητα τοποθέτησης του δείκτη του αρχείου κατευθείαν σε οποιοδήποτε στοιχείο.
3. Να υπάρχει η δυνατότητα να μάθουμε σε ποιο στοιχείο του αρχείου είναι τοποθετημένος ο δείκτης του αρχείου.

Υπολογισμός θέσης μίας εγγραφής



ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Γ' ΕΞΑΜΗΝΟ

Μέθοδοι (constructors) για τη δημιουργία ενός αρχείου `RandomAccessFile`

Μέθοδος	Περιγραφή
<code>RandomAccessFile(String file, String mode)</code>	Δημιουργεί ένα ρεύμα που αντιστοιχεί στο αρχείο κατ' ευθείαν πρόσβασης με όνομα <code>file</code> . Το ρεύμα αυτό μπορεί να είναι είτε ρεύμα εισόδου είτε ρεύμα εισόδου και εξόδου ταυτόχρονα, ανάλογα με την τιμή της παραμέτρου <code>mode</code> . Η <code>mode</code> μπορεί να έχει την τιμή "r" οπότε πρόκειται για ρεύμα εισόδου ή "rw" οπότε πρόκειται για ρεύμα εισόδου/εξόδου.
<code>RandomAccessFile(File file, String mode)</code>	Όπως παραπάνω με τη διαφορά ότι το αντικείμενο <code>file</code> έχει οριστεί προηγουμένως μέσω της κλάσης <code>File</code>

```
RandomAccessFile RAFile1 = new RandomAccessFile("myfile1.dat", "r");
```

```
RandomAccessFile RAFile2 = new RandomAccessFile("myfile2.dat", "rw");
```

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Γ' ΕΞΑΜΗΝΟ

Βασικές Μέθοδοι της Κλάσης `RandomAccessFile`
που αναφέρονται στην κατευθείαν πρόσβαση

Μέθοδος	Περιγραφή
<code>void seek(long pos)</code>	Τοποθετεί τον δείκτη του αρχείου ακριβώς πριν από το byte που αναφέρεται στη θέση <code>pos</code>
<code>long getFilePointer()</code>	Επιστρέφει τη θέση που δείχνει ο δείκτης του αρχείου
<code>int skipBytes(int number)</code>	Μετακινεί το δείκτη του αρχείου κατά τον αριθμό των bytes που αναφέρεται στη παράμετρο <code>number</code>

Μέθοδοι που υλοποιούν διαδικασίες εισόδου της Κλάσης `RandomAccessFile`

Μέθοδος	Περιγραφή
<code>long length()</code>	Επιστρέφει το μήκος του αρχείου σαν ένα αριθμό από bytes
<code>int read()</code>	Διαβάζει ένα byte από το αρχείο, το οποίο επιστρέφεται σαν ένας ακέραιος στο διάστημα 0-255
<code>int read(byte[] b)</code>	Διαβάζει δεδομένα μέχρι <code>b.length</code> bytes από το αρχείο και τα τοποθετεί στον πίνακα <code>b</code> . Επιστρέφει τον αριθμό των bytes που διάβασε ή -1 εάν συναντήσει τέλος αρχείου.
<code>boolean readBoolean()</code>	Διαβάζει και επιστρέφει ένα δεδομένο τύπου boolean από το αρχείο.
<code>byte readByte()</code>	Διαβάζει ένα byte από το αρχείο σαν μία προσημασμένη τιμή των 8 bits
<code>char readChar()</code>	Διαβάζει τα επόμενα 2 bytes από το αρχείο και τα επιστρέφει σαν έναν Unicode χαρακτήρα
<code>int readInt()</code>	Διαβάζει τα επόμενα 4 bytes από το αρχείο, τα οποία επιστρέφει σαν έναν ακέραιο προσημασμένο αριθμό των 32 bits.
<code>float readFloat()</code>	Διαβάζει τα επόμενα 4 bytes από το αρχείο, τα οποία επιστρέφει σαν έναν πραγματικό αριθμό τύπου float.

Μέθοδοι που υλοποιούν διαδικασίες εξόδου της Κλάσης `RandomAccessFile`

Μέθοδος	Περιγραφή
<code>void setLength(long newLength)</code>	Τοποθετεί το μήκος του αρχείου με <code>newLength</code> . Εάν το τρέχον μήκος του αρχείου είναι μεγαλύτερο από <code>newLength</code> το μέρος του αρχείου που απομένει αποκόπτεται. Εάν είναι μικρότερο από <code>newLength</code> τότε το αρχείο επεκτείνεται. Στη δεύτερη περίπτωση οι τιμές των bytes που προστίθενται είναι μη-ορισμένες.
<code>void write(int b)</code>	Γράφει το byte <code>b</code> στο αρχείο
<code>void write(byte[] b)</code>	Γράφει <code>b.length</code> bytes από τον πίνακα <code>b</code> στο αρχείο αρχίζοντας από την τρέχουσα θέση του δείκτη του αρχείου.
<code>void writeChar(int ch)</code>	Γράφει το χαρακτήρα <code>ch</code> σαν μία τιμή των 2 bytes, με το πιο σημαντικό byte να γράφεται πρώτο.
<code>void writeBoolean(boolean b)</code>	Γράφει την boolean τιμή <code>b</code> σαν μία τιμή του ενός byte στο αρχείο.
<code>void writeInt(int v)</code>	Γράφει τον ακέραιο <code>v</code> σαν μία τιμή των 4 bytes στο αρχείο.

Δημιουργία Αρχείου Κατ' Ευθείαν Πρόσβασης με φάρμακα

```
public class Medicine {

    private int Code;
    private String Name;
    private String Type;
    private double Price;

    public Medicine(int Code, String Name, String Type, Double Price){
        this.Code=Code;
        this.Name=Name;
        this.Type=Type;
        this.Price=Price;
    }

    public Medicine() {
        this(0, "", "", 0.0);
    }

    public void setCode(int Code){
        this.Code=Code;
    }
}
```

Δημιουργία Αρχείου Κατ' Ευθείαν Πρόσβασης με φάρμακα

```

public void setName(String Name) {
    this.Name=Name;
}

public void setType(String Type) {
    this.Type=Type;
}

public void setPrice(double Price) {
    this.Price=Price;
}

public int getCode() {
    return Code;
}

public String getName() {
    return Name;
}

public String getType() {
    return Type;
}

public double getPrice() {
    return Price;
}
}

```

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Γ' ΕΞΑΜΗΝΟ

Δημιουργία Αρχείου Κατ' Ευθείαν Πρόσβασης με φάρμακα

```

import java.io.*;

public class FileMedicine extends Medicine {

    public FileMedicine(int Code, String Name, String Type, double Price){
        super(Code, Name, Type, Price);
    }

    public FileMedicine(){
        super();
    }

    public FileMedicine read(RandomAccessFile File) throws IOException {
        setCode(File.readInt());
        setName(cutString10(File));
        setType(cutString10(File));
        setPrice(File.readDouble());
        return this;
    }
}

```

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Γ' ΕΞΑΜΗΝΟ

**Δημιουργία Αρχείου Κατ' Ευθείαν Πρόσβασης με φάρμακα
(ανάγνωση φαρμάκου)**

```

public FileMedicine read(RandomAccessFile File) throws IOException {
    setCode(File.readInt());
    setName(cutString10(File));
    setType(cutString10(File));
    setPrice(File.readDouble());
    return this;
}

private String cutString10(RandomAccessFile File) throws IOException {
    char stringName[] = new char[10];
    for (int i=0; i<stringName.length; i++)
        stringName[i]=File.readChar();
    return new String(stringName).replace('\0', ' ');
}

```

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Γ' ΕΞΑΜΗΝΟ

**Δημιουργία Αρχείου Κατ' Ευθείαν Πρόσβασης με φάρμακα
(εγγραφή φαρμάκου)**

```

public void write(RandomAccessFile File) throws IOException {
    File.writeInt(getCode());
    putString10(File, getName());
    putString10(File, getType());
    File.writeDouble(getPrice());
}

private void putString10(RandomAccessFile File, String S) throws IOException {
    StringBuffer buf=null;
    if (S!=null){
        buf=new StringBuffer(S);
        buf.setLength(10); }
    else
        buf=new StringBuffer(10);
}

```

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Γ' ΕΞΑΜΗΝΟ

Δημιουργία Αρχείου Κατ' Ευθείαν Πρόσβασης με φάρμακα**(εναλλακτικά: τοποθέτηση String μήκους 15)**

```
private void putString15(RandomAccessFile F, String S) throws IOException {
    if (S.length()==15)
        F.writeChars(S);
    else if (S.length() >15) {
        for(int i=0;i<15;i++)
            F.writeChar(S.charAt(i));
    }
    else {
        for(int i=0;i<S.length();i++)
            F.writeChar(S.charAt(i));
        for(int i=S.length();i<15;i++)
            F.writeChar(' ');
    }
}
```

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Γ' ΕΞΑΜΗΝΟ

Αρχείο Κατ' Ευθείαν Πρόσβασης με φοιτητές

```
public static void main(String[] args) {
    try {
        RandomAccessFile FTlout = new RandomAccessFile("DOMES","rw");
        /* Προτοποθέτηση αρχείου με 200 κενές εγγραφές φοιτητών
           Κάθε φοιτητής καταλαμβάνει 72 bytes στο αρχείο
           (4 bytes ο ΑΜ + 30 bytes το επίθετο + 30 bytes το όνομα +
            4 bytes οι απουσίες + 4 bytes ο βαθμός)
        */
        for (int i=0;i<200;i++) {
            Student S= new Student(170001+i);
            S.writeStudent(FTlout);
        }
        FTlout.close();
    }
}
```

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Γ' ΕΞΑΜΗΝΟ

Αρχείο Κατ' Ευθείαν Πρόσβασης με φοιτητές

```
// Νέο Άνοιγμα του αρχείου για επεξεργασία
RandomAccessFile FT1 = new RandomAccessFile("DOMES","rw");

// εγγραφή του φοιτητή με αριθμό μητρώου 170012
Student S1 = new Student(170012,"Papadopoulos", "Nikos");
FT1.seek((S1.getAm()-170001)*72);
S1.writeStudent(FT1);

// εγγραφή του φοιτητή με αριθμό μητρώου 170001
Student S2 = new Student(170001,"Papadakis", "Dimitris");
FT1.seek((S2.getAm()-170001)*72); // Αρχή του αρχείου (byte 0)
S2.writeStudent(FT1);

Student S = new Student();

// Εμφάνιση του φοιτητή με αριθμό μητρώου 170012 στην οθόνη
FT1.seek((17012-17001)*72);
S.readStudent(FT1);
System.out.println(S.toString());
```

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Γ' ΕΞΑΜΗΝΟ

Αρχείο Κατ' Ευθείαν Πρόσβασης με φοιτητές

```
// Εμφάνιση του φοιτητή με αριθμό μητρώου 170001 στην οθόνη
FT1.seek((170001-170001)*72); // Ο πρώτος φοιτητής του αρχείου
S.readStudent(FT1);
System.out.println(S.toString());

// Καταχώρηση απουσίας στο φοιτητή με αριθμό μητρώου 170012
FT1.seek((170012-170001)*72);
S.readStudent(FT1);
S.setAbsences(S.getAbsences()+1);
FT1.seek(FT1.getFilePointer()-72); // πήγαινε ξανά στην αρχή της εγγ
S.writeStudent(FT1);

// Εμφάνιση του φοιτητή με αριθμό μητρώου 170012 στην οθόνη
FT1.seek((12012-12001)*72);
S.readStudent(FT1);
System.out.println(S.toString());

}
catch (IOException e) {
    System.out.println("ERROR: " + e);}
}
```

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Γ' ΕΞΑΜΗΝΟ