Αρχεία - Ροές Δεδομένων Εισόδου και Εξόδου

Η κλάση File

Ενδεικτικοί κατασκευαστές και μέθοδοι

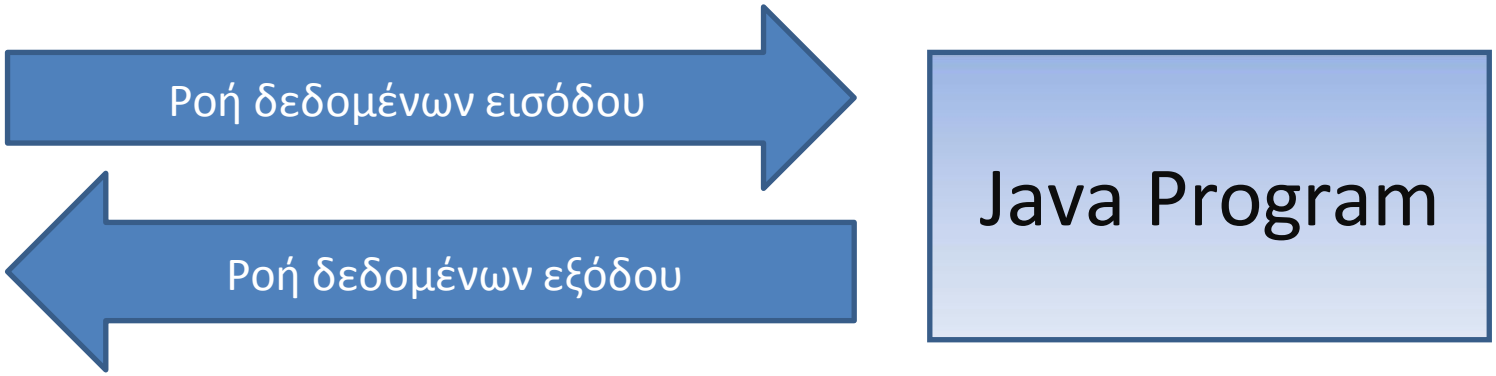
File(String pathname) Creates a new File instance by converting the given pathname string into an abstract pathname.

boolean	**canExecute**() Tests whether the application can execute the file
boolean	**canRead**() Tests whether the application can read the file
boolean	**canWrite**() Tests whether the application can modify the file
boolean	**delete**() Deletes the file or directory denoted by this file
boolean	**exists**() Tests whether the file or directory exists.
String	**getName**() Returns the name of the file or directory denoted by this abstract pathname.
String	**getParent**() Returns the pathname string of this abstract pathname's parent, or null if this pathname does not name a parent directory.
boolean	**isDirectory**() Tests whether the file is directory.
boolean	**isFile**() Tests whether the file is a normal file.
boolean	**isHidden**() Tests whether the file is a hidden file.
String[]	**list**() Returns an array of strings naming the files and directories in the directory
boolean	**mkdir**() Creates the directory.

Παράδειγμα χρήσης της File

```
/* @author Nick Z. Zacharis
*/
public class JavaFileExample {

    public static void main(String[] args) {
        File fp = new File("C:\\jdk7\\bin\\javac.exe");

        if(fp.exists() && fp.canExecute()){
            System.out.println(fp.getAbsolutePath() + " is executable");
        }

        File folder = new File("C:\\jdk7");
        String [] Contents = folder.list();
        for(String s : Contents) {
            System.out.println(s);
        }

        fp = new File("C:\\jdk7\\README.html");
        fp.renameTo(new File("C:\\jdk7\\read.html"));
        fp.delete();
    }
}
```

Input/Output Streams



Μια ροή δεδομένων μπορεί να αποτελείται είτε από bytes είτε από χαρακτήρες. Μια ροή μπορεί να κατευθύνεται από/προς ένα αρχείο, ένα άλλο πρόγραμμα, μια περιφερειακή συσκευή, μια δικτυακή σύνδεση, ένα πίνακα.

BYTE STREAMS - χρησιμοποιούνται από τα προγράμματα για να διαβάσουν ή και να γράψουν δεδομένα των 8-bits. Για την ανάγνωση χρησιμοποιείται η FileInputStream και για την εγγραφή η FileOutputStream, οι οποίες κληρονομούν την InputStream και OutputStream αντίστοιχα.

TEXT STREAMS - χρησιμοποιούνται από τα προγράμματα για να διαβάσουν ή και να γράψουν χαρακτήρες 16-bit. Για την ανάγνωση χρησιμοποιείται η FileReader και για την εγγραφή η FileWriter, οι οποίες κληρονομούν την Reader και Writer αντίστοιχα.

FileInputStream

Ενδεικτικοί Κατασκευαστές
FileInputStream(File obj) Δημιουργεί ένα FileInputStream ανοίγοντας σύνδεση με το File obj.
FileInputStream(String name) Δημιουργεί ένα FileInputStream ανοίγοντας σύνδεση με το αρχείο, το path του οποίου περιέχεται στο String name

Ενδεικτικές μέθοδοι της FileInputStream	
int	available() Επιστρέφει μια εκτίμηση για το πλήθος των bytes που μπορεί να διαβάσει από το ρεύμα εισόδου χωρίς να υπάρχει κώλυμα από την επόμενη κλήση μιας μεθόδου για αυτό το ρεύμα εισόδου.
void	close() κλείνει το ρεύμα εισόδου και απελευθερώνει τους δεσμευμένους πόρους.
int	read() Διαβάζει και επιστρέφει ένα byte από δεδομένα και -1 για το τέλος του αρχείου.
int	read(byte[] b) Διαβάζει μέχρι b.length bytes από δεδομένα και τα αποθηκεύει στο πίνακα b. Επιστρέφει το πλήθος των bytes που αποθήκευσε στο πίνακα.
int	read(byte[] b, int off, int len) Διαβάζει μέχρι len bytes από δεδομένα και τα αποθηκεύει στο πίνακα b. Το σημείο έναρξης αποθήκευσης των δεδομένων στο πίνακα b δηλώνεται στη παράμετρο off. Επιστρέφει το πλήθος των bytes που αποθήκευσε στο πίνακα.

FileOutputStream

Ενδεικτικοί Κατασκευαστές
FileOutputStream(File file) Δημιουργεί ένα FileOutputStream ανοίγοντας σύνδεση με το File obj.
FileOutputStream(String name) Δημιουργεί ένα FileOutputStream ανοίγοντας σύνδεση με το αρχείο, το path του οποίου περιέχεται στο String name
Και οι δύο κατασκευαστές δέχονται τη λογική παράμετρο append

Ενδεικτικές μέθοδοι της FileOutputStream	
void	close() κλείνει το ρεύμα εξόδου και απελευθερώνει τους δεσμευμένους πόρους.
void	**write**(int b) Γράφει ένα byte από δεδομένα.
void	**write**(byte[] b) Γράφει μέχρι b.length bytes από δεδομένα.
void	**write**(byte[] b, int off, int len) Γράφει μέχρι len bytes από δεδομένα από το πίνακα b, με σημείο έναρξης τη παράμετρο off.

Παράδειγμα αντιγραφής αρχείου

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class FileProject {
   void CopyBinaryFiles() throws IOException
   {
      FileInputStream in = null;
      FileOutputStream out = null;

      try {
         in = new FileInputStream("c:\\javaexamples\\input.pdf");
         out = new FileOutputStream("c:\\javaexamples\\output.pdf");
         int c;

         while ((c = in.read()) != -1) {
            out.write(c);
       }   }
      catch(java.io.IOException ex) {
         System.out.println("Error : " + ex.getMessage());
      }
      finally {
         if (in != null) {
               in.close();
         }
         if (out != null) {
            out.close();
   }    }}
```

```
byte [] b = new byte[1024] ;
while((c = in.read(b)) != -1) {
    out.write(b, 0, c);
}
```

Η μέθοδος read() διαβάζει ένα byte (8-bits) και επιστρέφει ένα int για να μπορεί να επιστρέψει και τη τιμή -1 στη περίπτωση που φτάσει στο τέλος του αρχείου.

FileReader

Ενδεικτικοί Κατασκευαστές
FileReader(File file) Δημιουργεί ένα FileReader ανοίγοντας σύνδεση με το File obj.
FileReader(String name) Δημιουργεί ένα FileReader ανοίγοντας σύνδεση με το αρχείο, το path του οποίου περιέχεται στο String name

Ενδεικτικές μέθοδοι της FileInputStream	
String	getEncoding() Επιστρέφει τη κωδικοποίηση που χρησιμοποιείται από το ρεύμα εισόδου.
void	close() κλείνει το ρεύμα εισόδου και απελευθερώνει τους δεσμευμένους πόρους.
int	read() Διαβάζει και επιστρέφει ένα χαρακτήρα και -1 για το τέλος του αρχείου.
int	read(char[] b, int off, int len) Διαβάζει μέχρι len χαρακτήρες και τους αποθηκεύει στο πίνακα b. Το σημείο έναρξης αποθήκευσης των δεδομένων στο πίνακα b δηλώνεται στη παράμετρο off. Επιστρέφει το πλήθος των χαρακτήρων που αποθήκευσε στο πίνακα.

FileWriter

Ενδεικτικοί Κατασκευαστές
FileWriter(File file) Δημιουργεί ένα FileWriter ανοίγοντας σύνδεση με το File obj.
FileWriter(String name) Δημιουργεί ένα FileWriter ανοίγοντας σύνδεση με το αρχείο, το path του οποίου περιέχεται στο String name
Και οι δύο κατασκευαστές δέχονται τη λογική παράμετρο append

Ενδεικτικές μέθοδοι της FileOutputStream	
void	close() κλείνει το ρεύμα εξόδου και απελευθερώνει τους δεσμευμένους πόρους.
void	**write**(int c) Γράφει ένα χαρακτήρα.
void	**write**(String b, int off, int len) Γράφει μέχρι len χαρακτήρες από το αλφαριθμητικό b, με σημείο έναρξης τη παράμετρο off.
void	**write**(char[] b, int off, int len) Γράφει μέχρι len χαρακτήρες από το πίνακα b, με σημείο έναρξης τη παράμετρο off.

Παράδειγμα αντιγραφής αρχείου

```
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

public class FileProject {
   void CopyTextFiles() throws IOException
   {
      FileReader in = null;
      FileWriter out = null;

      try {
         in = new FileReader ("c:\\javaexamples\\source.txt");
         out = new FileWriter ("c:\\javaexamples\\dest.txt");
         int c;

         while ((c = in.read()) != -1) {
            out.write(c);
       }   }
      catch(java.io.IOException ex) {
         System.out.println("Error : " + ex.getMessage());
      }
      finally {
         if (in != null) {
               in.close();
         }
         if (out != null) {
            out.close();
```

```
char[] b = new char[1024] ;
 while((c = in.read(b, 0,1024)) != -1) {
      out.write(b, 0, c);
 }
```

Η μέθοδος read() διαβάζει ένα χαρακτήρα (16 bits) και επιστρέφει ένα int για να μπορεί να επιστρέψει και τη τιμή -1 στη περίπτωση που φτάσει στο τέλος του αρχείου.

BufferedReader και PrintWriter

BufferedReader
Διαβάζει το κείμενο από ένα ρεύμα εισόδου χαρακτήρα χρησιμοποιώντας ενδιάμεση μνήμη (buffer) ,ώστε να εξασφαλίσει την αποτελεσματική ανάγνωση των χαρακτήρων, πινάκων, και γραμμών κειμένου.

Η χρήση του BufferedReader επιταχύνει τη διαδικασία της ανάγνωσης και για αυτό το λόγο κατασκευάζουμε αντικείμενα αυτού του τύπου, τα οποία αρχικοποιούνται με ρεύμα εισόδου είτε από FileReader ή InputStreamReader.

PrintWriter
Εκτυπώνει μορφοποιημένες αναπαραστάσεις αντικειμένων όπως String, int, float κ.λπ.σε ένα ρεύμα εξόδου για κείμενο. Αυτή η κλάση υλοποιεί όλες τις μεθόδους εκτύπωσης βρίσκονται στο PrintStream. Δεν περιλαμβάνει μεθόδους για τη σύνταξη των πρώτων bytes, για το οποίο πρόγραμμα θα πρέπει να χρησιμοποιούν μη κωδικοποιημένα ρεύματα byte.

Το αυτόματο άδειασμα (flush) όλων των χαρακτήρων προς το ρεύμα εξόδου γίνεται κάθε φορά που καλούμε τη μέθοδο print ή println και όχι όταν συναντήσει ένα χαρακτήρα νέας γραμμής.

Καθορισμένα ρεύματα εισόδου - εξόδου

Όπως όλες οι γλώσσες προγραμματισμού, έτσι και Java ορίζει ρεύματα εισόδου εξόδου από/προς :

System.in : Το προκαθορισμένο ρεύμα εισόδου για το πρόγραμμα και συνήθως χρησιμοποιείται το πληκτρολόγιο.

System.out : Το προκαθορισμένο ρεύμα εξόδου για το πρόγραμμα και συνήθως χρησιμοποιείται η οθόνη.

System.err : Το προκαθορισμένο ρεύμα εξόδου του προγράμματος για την εμφάνιση σφαλμάτων και συνήθως χρησιμοποιείται η οθόνη.

Ανάγνωση χαρακτήρων από το πληκτρολόγιο

```
import java.io.*;

public class ReadKeyBoard {
  public static void main(String args[]) throws IOException
  {
    InputStreamReader cin = null;

    try {
      cin = new InputStreamReader(System.in);
      System.out.println("Type any character to display or x to exit.");
      char c;
      do {
        c = (char) cin.read();
        System.out.print(c);
      } while(c != 'x');
    }finally {
      if (cin != null) {
        cin.close();
      }
    }
  }
}
```