



Οντοκεντρικός Προγραμματισμός

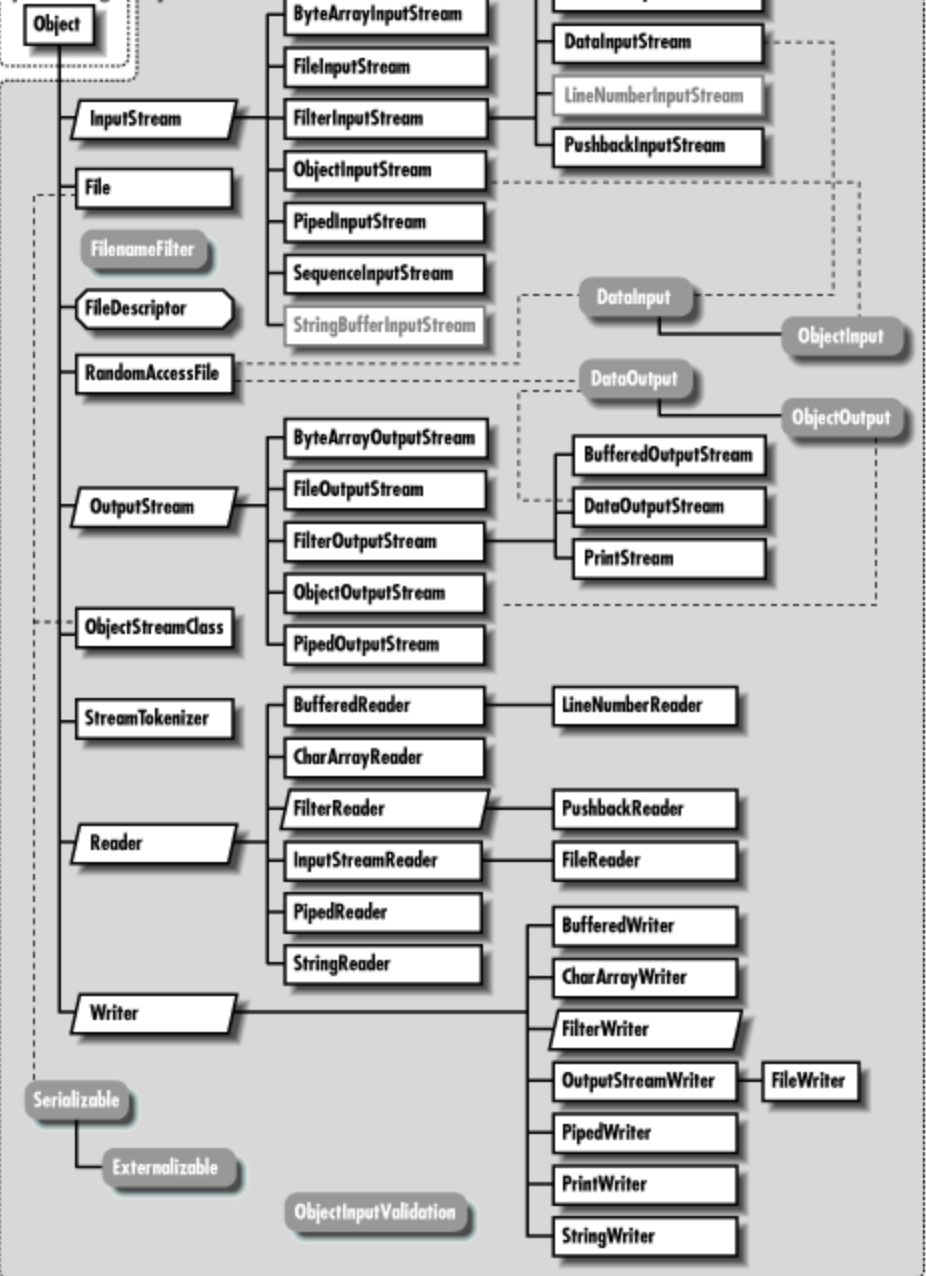
ΦΡΟΝΤΙΣΤΗΡΙΟ JAVA

Αρχεία δεδομένων

- Το πακέτο **java.io** περιλαμβάνει περισσότερες από 60 κλάσεις και διασυνδέσεις για το χειρισμό αρχείων δεδομένων.
- Αρκετές από τις κλάσεις του πακέτου `java.io` εντάσσονται σε μία από τις δύο κύριες κατηγορίες, οι οποίες διαχειρίζονται
 - αρχεία κειμένου
 - δυαδικά αρχεία

java.lang

java.io



KEY CLASS [rectangle] ABSTRACT CLASS [trapezoid] DEPRECATED CLASS [rectangle with diagonal line] FINAL CLASS [rounded rectangle] INTERFACE [oval]

--- implements --- extends

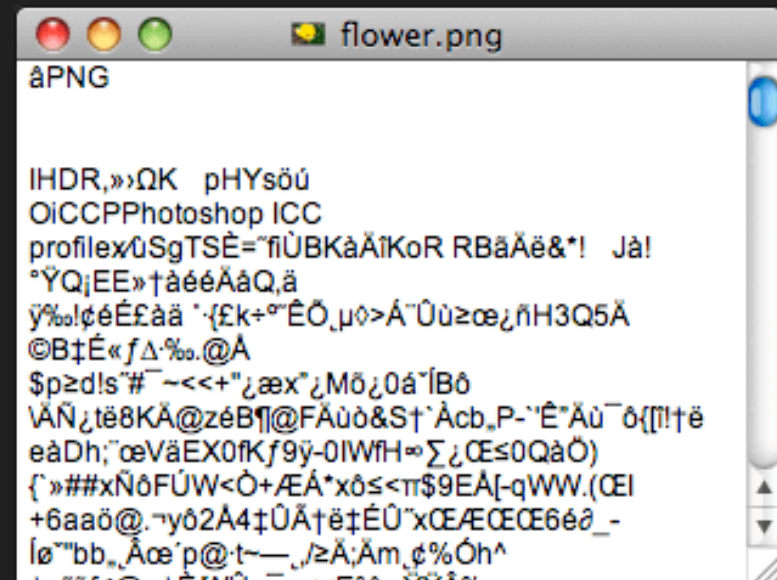
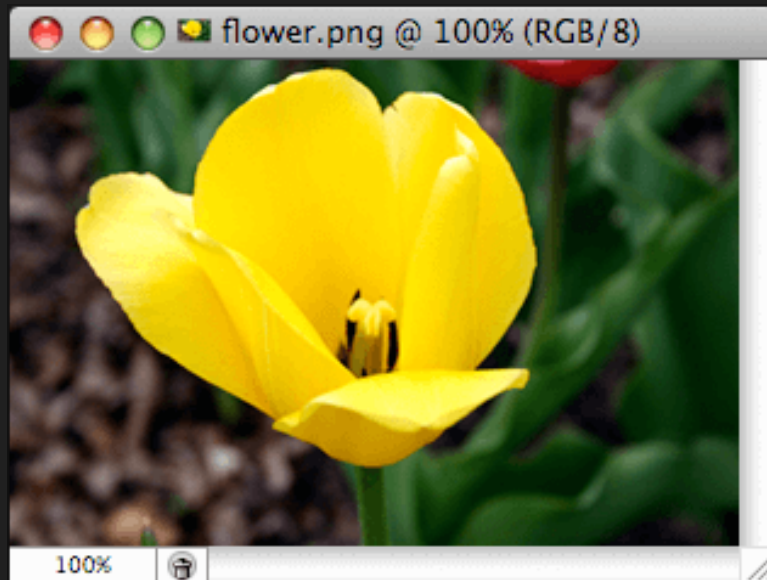
Αρχεία κειμένου

- Όταν μεταφέρουμε δεδομένα από ένα πρόγραμμα σε ένα αρχείο κειμένου, τότε όλα τα δεδομένα μετατρέπονται σε κείμενο πριν από την αποθήκευση.
 - Στην ουσία, τα αρχεία κειμένου περιλαμβάνουν δεδομένα σε μορφή παρόμοια με τον τύπο `char` της Java — συνήθως απλές, οργανωμένες σε γραμμές, αλφαριθμητικές πληροφορίες τις οποίες μπορεί να διαβάσει ο χρήστης χρησιμοποιώντας ένα συντάκτη κειμένου.
- Τα αρχεία κειμένου μπορούν να διαβαστούν και από προγράμματα που έχουν γραφεί σε άλλες γλώσσες προγραμματισμού, εκτός από την Java.

Δυαδικά αρχεία

- Όταν μεταφέρουμε δεδομένα από ένα πρόγραμμα σε ένα δυαδικό αρχείο, τότε τα δεδομένα δεν υφίστανται καμία μετατροπή πριν από την αποθήκευση.
- Τα δυαδικά αρχεία ποικίλουν περισσότερο: συνηθισμένο παράδειγμα είναι τα αρχεία εικόνων, και τα εκτελέσιμα προγράμματα.
- Σε ένα δυαδικό αρχείο μπορούμε να αποθηκεύσουμε με μία λειτουργία εγγραφής ακόμα και ένα ολόκληρο αντικείμενο ή συλλογή αντικειμένων, διαδικασία γνωστή ως σειριοποίηση.
- Ένα δυαδικό αρχείο μπορεί να διαβαστεί μόνο από προγράμματα όπου τα δεδομένα αναπαρίστανται εσωτερικά με τον ίδιο τρόπο όπως στο πρόγραμμα που δημιούργησε το συγκεκριμένο αρχείο.

Διαδικτά αρχεία



A screenshot of a web browser window showing the raw data of the flower image. The browser's address bar displays "flower.png". The content area shows the raw PNG data, starting with "âPNG" and followed by a series of hexadecimal characters representing the image's metadata and pixel data.

```
âPNG
IHDR,»»ΩK pHYsöú
oICCPPhotoshop ICC
profilex0SgTSE~"fiÛBKãÄiKoR RBãÄ&*! Jà!
°ÿQ;EE»†àééÄáQ,ä
ÿ%!çéÉ£ää`·{£k+°"ÉÖ,μφ>Á"ÛùzœζñH3Q5Ä
©B†É«fΔ·%o.@Ä
$Pzd!s"#~<<+"ζæx"ζMöζ0á"ÍBó
VÄÑζtë8KÄ@zëB¶@FÄùò&S†·Äcb_P-·"É"Äù"ó{¶†ë
eàDh;·œVäEX0fKf9ÿ-0IWfH∞Σζœ≤0QàÖ)
{`»##xÑòFÚW<Ö+ÆÁ*xò≤<π$9EA[-qWW.(œI
+6aaö@.·yò2A4†ÛÄ†ë†ÉÛ"xœÆœœœ6éθ_-
Íø™bb,·Äœ'p@:t—,./zÄ;Äm,ç%Óh^
```

Αρχεία δεδομένων

- Οι κλάσεις που αναφέρονται στη διαχείριση αρχείων κειμένου ονομάζονται αναγνώστες (readers) και γραφείς (writers).
- Οι κλάσεις που αναφέρονται στη διαχείριση δυαδικών αρχείων είναι γνωστές ως χειριστές ρευμάτων (stream).
- Σε αρκετές περιπτώσεις, ωστόσο, χρησιμοποιείται ο όρος ρεύμα δεδομένων είτε αναφερόμαστε σε είσοδο/έξοδο από/σε δυαδικά αρχεία είτε σε αρχεία κειμένου. Στη δεύτερη περίπτωση αναφερόμαστε σε ρεύματα κειμένου.

Αντικείμενα αρχείων – η κλάση File

- Για τη δημιουργία αντικειμένων αρχείων χρησιμοποιούμε την **κλάση File**.
- Τα **αντικείμενα File** χρησιμοποιούνται για να πάρουμε πληροφορίες για **φακέλους ή αρχεία**.
- Τα αντικείμενα File δεν χρησιμοποιούνται για την ανάγνωση ή εγγραφή σε φακέλους ή αρχεία.
 - Γι' αυτό το λόγο χρησιμοποιούνται τα ρεύματα εισόδου/εξόδου.

Αντικείμενα αρχείων – η κλάση File

Μέθοδος	Περιγραφή
<code>exists()</code>	Επιστρέφει <code>true</code> αν υπάρχει το αντικείμενο αρχείου ή φακέλου, και <code>false</code> διαφορετικά
<code>isDirectory()</code>	Επιστρέφει <code>true</code> αν το αντικείμενο αναφέρεται σε φάκελο, και <code>false</code> διαφορετικά
<code>isFile()</code>	Επιστρέφει <code>true</code> αν το αντικείμενο αναφέρεται σε αρχείο, και <code>false</code> διαφορετικά
<code>isAbsolute()</code>	Επιστρέφει <code>true</code> αν το αντικείμενο αναφέρεται σε μια πλήρη διαδρομή, και <code>false</code> διαφορετικά
<code>canRead()</code>	Επιστρέφει <code>true</code> αν μπορούμε να διαβάσουμε από το αρχείο, και <code>false</code> διαφορετικά
<code>canWrite()</code>	Επιστρέφει <code>true</code> αν μπορούμε να γράψουμε στο αρχείο, και <code>false</code> διαφορετικά
<code>getName()</code>	Επιστρέφει ένα αλφαριθμητικό, το οποίο αντιπροσωπεύει το όνομα του αρχείου ή το όνομα του φακέλου ανάλογα, χωρίς να αναφέρεται η διαδρομή
<code>getPath()</code>	Επιστρέφει ένα αλφαριθμητικό με το όνομα της διαδρομής
<code>length()</code>	Επιστρέφει έναν ακέραιο τύπου <code>long</code> που εκφράζει το μήκος του αρχείου σε <code>bytes</code>

Η κλάση File

Για τη δημιουργία ενός αντικειμένου File:

- Δημιουργούμε πρώτα ένα αντικείμενο που αντιστοιχεί στη διαδρομή του αρχείου (path):

```
File myDir = new File("C:\\projects\\files");
```

Το "\\\" είναι ο χαρακτήρας διαφυγής "\\\" μαζί με το διαχωριστικό διαδρομής "\\\"

- Στη συνέχεια, δημιουργούμε ένα αντικείμενο αρχείου που αντιστοιχεί στο ίδιο το αρχείο data.txt:

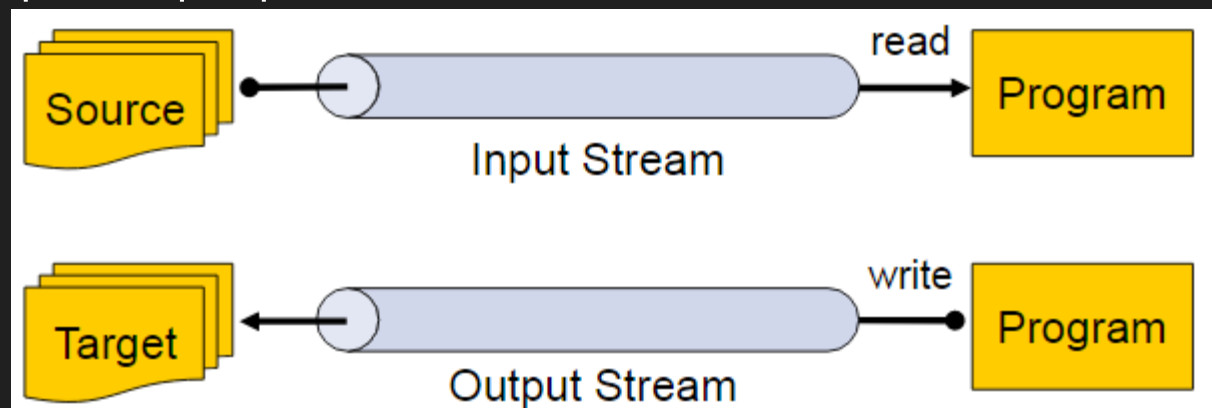
```
File myFile = new File(myDir, "data.txt");
```

- Μπορούμε να συνδυάσουμε τις δύο εργασίες ως εξής:

```
File myFile = new File("C:\\projects\\files", "data.txt");
```

Ρεύματα δεδομένων

- Τα δεδομένα που ανταλλάσσονται μεταξύ του προγράμματος και ενός αρχείου ως ένα ρεύμα.
- Τα δεδομένα ρέουν από μια προέλευση (source) προς ένα προορισμό (destination).
- Αρχικά, «ανοίγουμε» το ρεύμα.
- Αν το ρεύμα ρέει από το πρόγραμμα προς το αρχείο, τότε «γράφουμε» στο ρεύμα.
- Αν το ρεύμα ρέει από αρχείο προς το πρόγραμμα, τότε το πρόγραμμα «διαβάζει» δεδομένα.
- Τέλος, «κλείνουμε» το ρεύμα.



Ρεύματα δεδομένων

Χρησιμοποιώντας την καθιερωμένη ορολογία, διακρίνουμε τα εξής βήματα:

- **Άνοιγμα αρχείου:** αντιστοιχίζουμε ένα αντικείμενο ρεύματος στο αρχείο από το οποίο θέλουμε να διαβάσουμε ή στο οποίο θέλουμε να γράψουμε.
- **Εγγραφή/ανάγνωση:** πραγματοποιείται η επικοινωνία με το φυσικό αρχείο δεδομένων διαβιβάζοντας μηνύματα εγγραφής και ανάγνωσης στο αντικείμενο ρεύματος που έχουμε δημιουργήσει. Στην ουσία, το φυσικό αρχείο δεδομένων «κρύβεται» μέσα στο αντικείμενο ρεύματος.
- **Κλείσιμο αρχείου:** κλείσιμο του αντικειμένου ρεύματος και τερματισμός της σύνδεσης με το φυσικό αρχείο δεδομένων. Ταυτόχρονα, απελευθερώνονται οι εσωτερικοί πόροι που είχαν δοθεί στο αντικείμενο ρεύματος. **Αν ξεχάσουμε να κλείσουμε ένα αρχείο στο οποίο γράφει το πρόγραμμα, τότε υπάρχει κίνδυνος απώλειας δεδομένων.**

Ρεύματα δεδομένων (3)

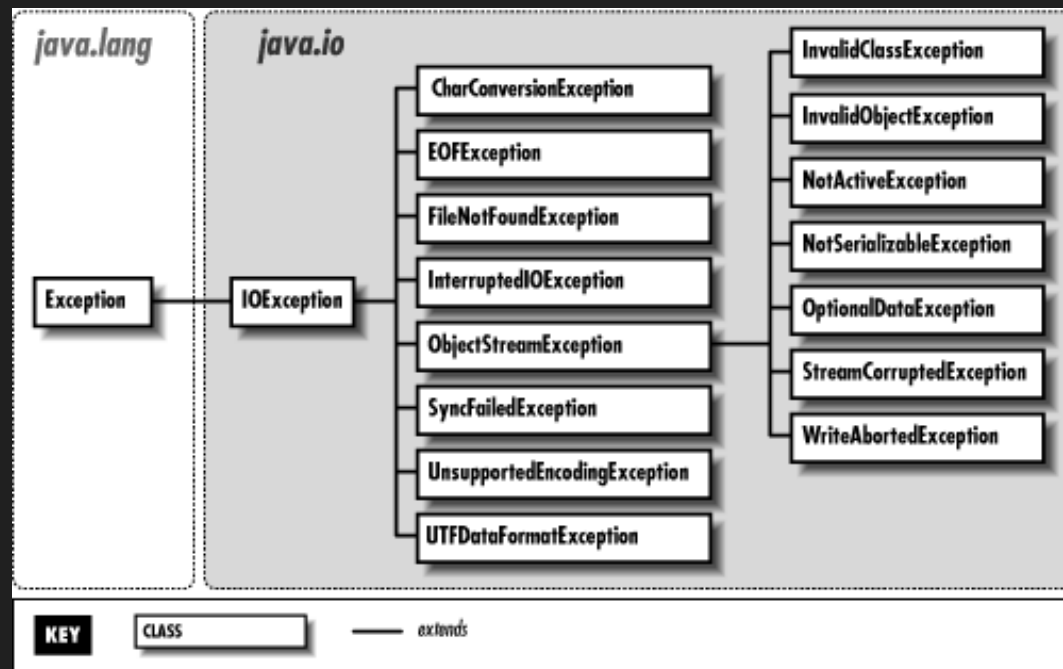
- Όπως ήδη αναφέρθηκε, υπάρχουν πολλές κλάσεις στο πακέτο `java.io` για τη δημιουργία αντικειμένων ρεύματος.
- Η δημιουργία ενός αντικειμένου ρεύματος γίνεται σε αρκετές περιπτώσεις με έναν ιδιαίτερο τρόπο: **ένα στιγμιότυπο μιας κλάσης ρεύματος μπορεί να χρησιμοποιηθεί ως όρισμα στον κατασκευαστή μιας άλλης κλάσης ρεύματος**. Η διαδικασία αυτή μπορεί να επαναληφθεί αρκετές φορές μέχρι να καταλήξουμε σε ένα ρεύμα που θα μας ικανοποιεί.

Εγγραφή σε αρχείο κειμένου: η κλάση FileWriter ⁽¹⁾

- Η αποθήκευση δεδομένων σε αρχείο γίνεται, όπως ήδη αναφέρθηκε, σε τρία βήματα:
 1. Άνοιγμα του αρχείου
 2. Εγγραφή των δεδομένων
 3. Κλείσιμο του αρχείου
- Οποιοδήποτε από αυτά τα βήματα μπορεί να αποτύχει για διαφορετικούς λόγους, πολλοί από τους οποίους είναι πέρα από τον έλεγχο του προγραμματιστή της εφαρμογής. Για παράδειγμα, μπορεί να υπάρχει αδυναμία ανοίγματος του αρχείου ή εγγραφής στο σκληρό δίσκο κτλ.

Εγγραφή σε αρχείο κειμένου: η κλάση FileWriter (1)

- Συνεπώς, πρέπει να αναμένουμε την παραγωγή **εξαιρέσεων** σε οποιοδήποτε από τα παραπάνω στάδια.



Εγγραφή σε αρχείο κειμένου: η κλάση `FileWriter` (2)

Βήμα 1^ο: άνοιγμα αρχείου

- Δημιουργούμε ένα αντικείμενο `FileWriter`, του οποίου ο κατασκευαστής δέχεται το όνομα του αρχείου.

```
FileWriter writer = new FileWriter("...όνομα  
αρχείου...");
```

- Αν το αρχείο δεν υπάρχει ήδη, δημιουργείται.
- Αν το αρχείο υπάρχει και θέλουμε να γίνει εγγραφή μετά από όσα υπάρχουν ήδη στο αρχείο, τότε χρησιμοποιούμε τον παρακάτω κατασκευαστή

```
public FileWriter(String filename, boolean append)
```

με δεύτερο όρισμα την τιμή `true`.

- Το όνομα του αρχείου μπορεί να έχει τη μορφή αλφαριθμητικού ή ενός αντικειμένου `File`:

```
File f = new File("...όνομα αρχείου...");
```


Εγγραφή σε αρχείο κειμένου: η κλάση `FileWriter` (3)

Βήμα 1^ο: άνοιγμα αρχείου

- η δημιουργία ενός αντικειμένου `FileWriter` έχει ως αποτέλεσμα το άνοιγμα του εξωτερικού αρχείου και την προετοιμασία του για τη λήψη της εξόδου (=εγγραφή).
- Αν το άνοιγμα του αρχείου αποτύχει για οποιονδήποτε λόγο, τότε ο κατασκευαστής θα παραγάγει μια εξαίρεση `IOException`.

Εγγραφή σε αρχείο κειμένου: η κλάση FileWriter (4)

Βήμα 2^ο: εγγραφή δεδομένων

- Όταν ένα αρχείο ανοίξει επιτυχώς, τότε μπορεί να χρησιμοποιηθεί η μέθοδος `write` του εγγραφέα για την αποθήκευση χαρακτήρων — συχνά με μορφή αλφαριθμητικών — στο αρχείο:

```
writer.write("...τμήμα κειμένου...");
```

- Ακόμα και αν το αρχείο ανοίξει επιτυχώς, υπάρχει πιθανότητα να αποτύχει οποιαδήποτε προσπάθεια εγγραφής σε αυτό.

Εγγραφή σε αρχείο κειμένου: η κλάση FileWriter (5)

Βήμα 3^ο: κλείσιμο αρχείου

- Όταν τελειώσει η εγγραφή της εξόδου, πρέπει να κλείσουμε κανονικά το αρχείο:

```
writer.close();
```

- Αυτό εξασφαλίζει ότι όλα τα δεδομένα έχουν πράγματι γραφεί στο εξωτερικό σύστημα αρχείων, και έχει συχνά ως αποτέλεσμα την αποδέσμευση κάποιων εσωτερικών ή εξωτερικών πόρων.
- Πάλι, υπάρχει πιθανότητα (αν και μικρή) να αποτύχει η προσπάθεια να κλείσουμε το αρχείο.

Εγγραφή σε αρχείο κειμένου: η κλάση FileWriter

```
File f = new File("... όνομα αρχείου ...");  
FileWriter writer = new FileWriter(f);  
while(υπάρχει επιπλέον κείμενο για εγγραφή)  
{  
    ...  
    writer.write(επόμενο τμήμα κειμένου);  
    ...  
}  
writer.close();
```

Εγγραφή σε αρχείο κειμένου: η κλάση FileWriter

```
try
{
    File f = new File("... όνομα αρχείου ...");
    FileWriter writer = new FileWriter(f);
    while(υπάρχει επιπλέον κείμενο για εγγραφή)
    {
        ...
        writer.write("επόμενο τμήμα κειμένου");
        ...
    }
    writer.close();
}
catch(IOException e)
{
    προέκυψε κάποιο λάθος στην προσπέλαση του αρχείου
}
```

Ανάγνωση από αρχείο κειμένου: η κλάση `FileReader` ⁽¹⁾

- Το διάβασμα δεδομένων από αρχείο γίνεται, όπως ήδη αναφέρθηκε, σε τρία βήματα:
 1. Άνοιγμα του αρχείου
 2. Ανάγνωση των δεδομένων
 3. Κλείσιμο του αρχείου

Ανάγνωση από αρχείο κειμένου: η κλάση `FileReader` (2)

- Ενώ οι φυσικές μονάδες για την εγγραφή κειμένου είναι οι χαρακτήρες και τα αλφαριθμητικά, οι αντίστοιχες για την ανάγνωση του κειμένου είναι οι **χαρακτήρες** και οι **γραμμές**.
- Ωστόσο, παρόλο που η κλάση `FileReader` περιλαμβάνει μια μέθοδο για την ανάγνωση ενός χαρακτήρα, **δεν περιλαμβάνει κάποια μέθοδο για την ανάγνωση μίας γραμμής**.
- Το πρόβλημα με την ανάγνωση γραμμών από αρχείο είναι ότι δεν υπάρχει κάποιο προκαθορισμένο όριο για το μήκος γραμμής.
- Αυτό σημαίνει ότι οποιαδήποτε μέθοδος χρησιμοποιηθεί για την επιστροφή της επόμενης ολόκληρης γραμμής από ένα αρχείο πρέπει να μπορεί να διαβάσει έναν οσοδήποτε μεγάλο αριθμό χαρακτήρων.

Ανάγνωση από αρχείο κειμένου: η κλάση `FileReader` ⁽³⁾

- Γι' αυτό, συνήθως, το αντικείμενο `FileReader` περιτυλίγεται σε ένα αντικείμενο `BufferedReader`:

```
FileReader freader = new FileReader("../όνομα  
αρχείου...");
```

```
BufferedReader reader = new BufferedReader(freader);
```

ή συνοπτικά

```
BufferedReader reader = new BufferedReader(  
    new FileReader("../όνομα αρχείου..."));
```

- Ένα αντικείμενο `BufferedReader` διαθέτει τη μέθοδο `readLine` για το διάβασμα μιας γραμμής.
- Ο χαρακτήρας τερματισμού γραμμής αφαιρείται πάντα από το αλφαριθμητικό που επιστρέφεται
- Το τέλος ενός αρχείου υποδηλώνεται με μια τιμή `null`.

Ανάγνωση από αρχείο κειμένου: η κλάση `FileReader` (4)

- Η κλάση `BufferedReader`, εκτός από το γεγονός ότι διαθέτει τη μέθοδο `readLine`, παρουσιάζει και ένα άλλο σημαντικό πλεονέκτημα:
- Η κλάση `BufferedReader` χρησιμοποιεί μια **περιοχή προσωρινής αποθήκευσης (buffer)**, δηλαδή μια περιοχή στη μνήμη που λειτουργεί ως προσωρινός αποθηκευτικός χώρος.
- Το τυπικό μέγεθος μιας περιοχής προσωρινής αποθήκευσης που χρησιμοποιείται για την ενδιάμεση αποθήκευση δεδομένων από ένα αρχείο σε ένα πρόγραμμα Java είναι **8192 χαρακτήρες**.
- Με τη χρήση του `buffer` η προσπάθεια του δίσκου είναι λιγότερο συχνή, κάτι που αυξάνει την ταχύτητα του προγράμματος.

Ανάγνωση από αρχείο κειμένου: η κλάση `FileReader` (5)

```
File f = new File("... όνομα αρχείου ...");
BufferedReader reader = new BufferedReader(new FileReader(f));
String line = reader.readLine();
while(line != null)
{
    κάποια ενέργεια με τη γραμμή
    line = reader.readLine();
}
reader.close();
```

Είσοδος κειμένου: η κλάση FileReader (5)

```
try
{
    File f = new File("... όνομα αρχείου ...");
    BufferedReader reader = new BufferedReader(new FileReader(f));
    String line = reader.readLine();
    while(line != null)
    {
        κάποια ενέργεια με τη γραμμή
        line = reader.readLine();
    }
    reader.close();
}
catch(FileNotFoundException e)
{
    δεν εντοπίστηκε το συγκεκριμένο αρχείο
}
catch(IOException e)
{
    κάποιο πρόβλημα προέκυψε στην ανάγνωση ή στο κλείσιμο
}
```

Παράδειγμα εγγραφής σε αρχείο κειμένου

- `fileWriter` →
- `fileReader` →

Σειριοποίηση αντικειμένων (1)

- Η **σειριοποίηση** επιτρέπει
 - την εγγραφή ενός ολόκληρου αντικειμένου σε ένα εξωτερικό αρχείο με μία λειτουργία
 - την ανάγνωση ενός ολόκληρου αντικειμένου από ένα εξωτερικό αρχείο με μία λειτουργία
- Η σειριοποίηση επιτρέπει επίσης να γραφούν και να διαβαστούν αντικείμενα όχι μόνο σε ένα σύστημα αρχείων, αλλά και σε ένα **δίκτυο**.
- Η σειριοποίηση λειτουργεί τόσο με απλά αντικείμενα όσο και με αντικείμενα πολλών στοιχείων, όπως οι **συλλογές**.
- Με την σειριοποίηση αποφεύγεται η ανάγνωση και η εγγραφή αντικειμένων πεδίο προς πεδίο.

Σειριοποίηση αντικειμένων (2)

- Προκειμένου να είναι κατάλληλη για σειριοποίηση, μια κλάση πρέπει να υλοποιεί τη διασύνδεση `Serializable` η οποία ορίζεται στο πακέτο `java.io`.
- Η διασύνδεση `Serializable` δεν ορίζει μεθόδους (`marking interface`).
- Αυτό σημαίνει ότι γίνεται αυτόματη διαχείριση της διαδικασίας σειριοποίησης από το σύστημα χρόνου εκτέλεσης, και από το χρήστη απαιτείται απλά να γράψει λίγες γραμμές κώδικα.
- Αν κάποιο πεδίο της κλάσης που έχουμε δηλώσει ότι υλοποιεί το `Serializable` interface δεν θέλουμε να γίνει `serialize` (δηλ. να αποθηκεύεται) τότε το δηλώνουμε σαν `transient`.

Εγγραφή σε δυαδικό αρχείο (1)

- Η αποθήκευση αντικειμένων σε αρχείο γίνεται σε τρία βήματα:
 1. Άνοιγμα του αρχείου
 2. Εγγραφή των δεδομένων
 3. Κλείσιμο του αρχείου
- Κάθε αντικείμενο που εμπλέκεται πρέπει να προέρχεται από μια κλάση που υλοποιεί τη διασύνδεση [Serializable](#).
- Για να γράψουμε αντικείμενα σε ένα αρχείο πρέπει να δημιουργήσουμε ένα αντικείμενο [ObjectOutputStream](#), ακολουθώντας τα παρακάτω βήματα:

Εγγραφή σε δυαδικό αρχείο (2)

Βήμα 1^ο: άνοιγμα αρχείου για εγγραφή αντικειμένων

1. Δημιουργούμε ένα αντικείμενο αρχείου `File`:

```
File f = new File(".....");
```

2. Δημιουργούμε ένα αντικείμενο `FileOutputStream` χρησιμοποιώντας ως όρισμα το αντικείμενο `File` που δημιουργήσαμε παραπάνω:

```
FileOutputStream fouts = new FileOutputStream(f);
```

3. Δημιουργούμε ένα αντικείμενο `ObjectOutputStream` χρησιμοποιώντας ως όρισμα το αντικείμενο `fouts`

```
ObjectOutputStream douts = new ObjectOutputStream(fouts);
```

Οι δύο τελευταίες εντολές μπορούν να συνδυαστούν σε μία:

```
ObjectOutputStream douts = new ObjectOutputStream(  
new FileOutputStream(f));
```


Έξοδος σε δυαδικό αρχείο (3)

Βήμα 2^ο: εγγραφή αντικειμένων

- Για την εγγραφή των αντικειμένων χρησιμοποιείται η μέθοδος `writeObject()`:

```
douts.writeObject(αντικείμενο);
```

Βήμα 3^ο: κλείσιμο αρχείου

```
douts.close();
```

```
fouts.close();
```

Παράδειγμα: `Serializ1`



Ανάγνωση από δυαδικό αρχείο ⁽¹⁾

- Η ανάγνωση αντικειμένων από αρχείο γίνεται σε τρία βήματα:
 1. Άνοιγμα του αρχείου
 2. Ανάγνωση των αντικειμένων
 3. Κλείσιμο του αρχείου
- Κάθε αντικείμενο που εμπλέκεται πρέπει να προέρχεται από μια κλάση που υλοποιεί τη διασύνδεση **Serializable**.
- Για να διαβάσουμε αντικείμενα από ένα αρχείο πρέπει να δημιουργήσουμε ένα αντικείμενο **ObjectInputStream**, ακολουθώντας τα παρακάτω βήματα:

Ανάγνωση από δυαδικό αρχείο (2)

Βήμα 1^ο: άνοιγμα αρχείου για διάβασμα αντικειμένων

1. Δημιουργούμε ένα αντικείμενο αρχείου `File`:

```
File f = new File("...");
```

2. Δημιουργούμε ένα αντικείμενο `FileInputStream` χρησιμοποιώντας ως όρισμα το αντικείμενο `File` που δημιουργήσαμε παραπάνω:

```
FileInputStream fins = new  
FileInputStream(f);
```

3. Δημιουργούμε ένα αντικείμενο `ObjectInputStream` χρησιμοποιώντας ως όρισμα το αντικείμενο `fins`

```
ObjectInputStream dins = new ObjectInputStream(fins);
```

Οι δύο τελευταίες εντολές μπορούν να συνδυαστούν σε μία:

```
ObjectInputStream dins =  
new ObjectInputStream(new FileInputStream(f));
```

Ανάγνωση από δυαδικό αρχείο (3)

Βήμα 2ο: διάβασμα αντικειμένων

- Για την ανάγνωση/διάβασμα των αντικειμένων χρησιμοποιείται η μέθοδος `readObject` της κλάσης `ObjectInputStream`.
- Η μέθοδος `readObject` επιστρέφει ένα αντικείμενο τύπου `Object`, οπότε χρειάζεται να γίνει `casting` για την προσαρμογή του επιστρεφόμενου αντικειμένου στον κατάλληλο τύπο:

```
(τύπος αντικειμένου) dins.readObject();
```

- **Βήμα 3ο: κλείσιμο αρχείου**

```
dins.close();
```

```
fins.close();
```

Παράδειγμα: `Serializ2`

