

10 ΕΙΣΟΔΟΣ - ΕΞΟΔΟΣ ΔΕΔΟΜΕΝΩΝ

Οι περισσότερες εφαρμογές της Java δε στηρίζονται σε προγράμματα, τα οποία δέχονται είσοδο δεδομένων απ' το πληκτρολόγιο, αλλά σε applets, εφαρμογές που βασίζονται στην παραθυρική επικοινωνία με το χρήστη. Γι' αυτό το λόγο, το μέρος του συστήματος Εισόδου/Εξόδου της Java (I/O, Input/Output System) δε χρησιμοποιείται πολύ στον εμπορικό κώδικα, αλλά περισσότερο για διδακτικούς λόγους. Το Σύστημα Εισόδου/Εξόδου της Java στηρίζεται στα streams δεδομένων (**δέσμες** ή **ρεύματα** δεδομένων), τα οποία συνδέονται με μια φυσική συσκευή, πληκτρολόγιο, οθόνη ή αρχείο. Υπάρχουν δύο τύποι streams, byte και χαρακτήρων. Τα streams χαρακτήρων προστέθηκαν στις τελευταίες εκδόσεις της Java. Τα byte streams παρέχουν έναν εύκολο τρόπο στο χειρισμό δεδομένων και είναι πολύ αποτελεσματικά στην ανάγνωση και εγγραφή δυαδικών δεδομένων από και σε αρχεία. Τα streams χαρακτήρων σχεδιάστηκαν για το χειρισμό εισόδου και εξόδου χαρακτήρων. Στηρίζονται στα byte streams, οπότε μετατρέπουν τα bytes σε χαρακτήρες. Αν και το σύστημα Εισόδου/Εξόδου της Java περιλαμβάνει πολλές και διάφορες κλάσεις και υποκλάσεις, ο απλός χρήστης που δεν έχει ακόμη γνώση της κληρονομικότητας θα πρέπει απλώς να γνωρίζει ότι η εντολή `System.out.println()` για παράδειγμα είναι η κλήση της μεθόδου `println()` του αντικείμενου `System.out`, το οποίο αναφέρεται στο στάνταρ stream εξόδου, την οθόνη. Παρόμοια, το αντικείμενο `System.in`, αναφέρεται στο στάνταρ stream εισόδου, το πληκτρολόγιο.

10.1 Byte Streams

Οι κλάσεις που χρησιμοποιούνται για είσοδο και έξοδο δεσμών bytes (byte streams) είναι οι κλάσεις `InputStream` και `OutputStream` αντίστοιχα. Το `System.in` είναι αντικείμενο τύπου `InputStream`, ενώ το `System.out` είναι αντικείμενο τύπου `OutputStream`. Τα αντικείμενα αυτά συνδέονται και με τις αντίστοιχες μεθόδους. Έτσι, η μορφή που μπορεί να πάρει η κλήση της μεθόδου για διάβασμα χαρακτήρων σε μορφή bytes είναι :

```
int read() : Επιστρέφει σε ακέραιο το περιεχόμενο ενός byte, χρειάζεται διανομή.  
int read(byte b[]) : Επιστρέφει τον αριθμό των bytes που διάβασε με επιτυχία στον  
πίνακα b[] από bytes  
int read(byte b[], int offset, int numofBytes) : Διαβάζει έναν πίνακα  
από bytes από το b[offset] μέχρι το numofBytes
```

ενώ για την εμφάνιση των χαρακτήρων – bytes χρησιμοποιούνται οι μέθοδοι :

```
void write(int b) : Εμφανίζει ένα χαρακτήρα  
void write(byte b[]) : Εμφανίζει έναν πίνακα από bytes
```

`void write(byte b[], int offset, int numBytes) :` Εμφανίζει όσους χαρακτήρες επιλέξουμε από έναν πίνακα από `bytes`

Παράδειγμα 1

Με την εντολή

```
char ch = (char) System.in.read();
```

διαβάζουμε ένα `byte`, το μετατρέπουμε σε χαρακτήρα με **διανομή** (`char`) και αποθηκεύουμε το χαρακτήρα στη μεταβλητή `ch`.

Παράδειγμα 2

Με το επόμενο πρόγραμμα δηλώνουμε έναν πίνακα από `bytes`, τον `bytesPin[]`, στον οποίο αποθηκεύουμε ένα όνομα, το οποίο εμφανίζουμε με την εντολή `System.out.write()` :

```
public class ReadBytePin {  
  
    public static void main(String[] args) {  
        int numBytes = 0;  
        byte bytesPin[] = new byte[100];  
        System.out.print("Enter Your Name : ");  
  
        try {  
            numBytes = System.in.read(bytesPin);  
            System.out.println("You Entered " + numBytes + " Characters");  
            System.out.print("Your Name Is : ");  
            System.out.write(bytesPin, 0, numBytes);  
            System.out.println();  
        }  
  
        catch (java.io.IOException e) {  
            System.out.println("Error Reading Your Name");  
        }  
    }  
}
```

Έξοδος Προγράμματος

```
run:  
Enter Your Name : Kostas Goulianas  
You Entered 17 Characters  
Your Name Is : Kostas Goulianas  
BUILD SUCCESSFUL (total time: 2 seconds)
```

Παρατήρηση

Τα στοιχεία του πίνακα `bytesPin` μπορούν να εμφανιστούν και με την εντολή `System.out.print()` και `System.out.println()`.

10.2 Streams Χαρακτήρων

Το `System.in` (η είσοδος δεδομένων δηλαδή) είναι αντικείμενο τύπου `InputStream`, μιας κλάσης που χρησιμοποιείται για είσοδο και έξοδο δεσμών `bytes` (`byte streams`). Για να χρησιμοποιήσουμε είσοδο και έξοδο δεδομένων που βασίζονται σε `streams` χαρακτήρων, θα πρέπει να χρησιμοποιήσουμε τις αντίστοιχες κλάσεις που βασίζονται σε `streams` χαρακτήρων, έτσι ώστε το `System.in` που είναι `byte stream` να μετατραπεί σε `stream` χαρακτήρων. Αυτό γίνεται με τη δημιουργία ενός αντικειμένου τύπου `InputStreamReader`, όπου το `System.in` περνάει σαν παράμετρος. Αυτό το νέο αντικείμενο περνάει σαν παράμετρος στη δημιουργία ενός αντικειμένου ενδιάμεσης μνήμης τύπου `BufferedReader`. Πιο αναλυτικά :

Με την εντολή :

```
InputStreamReader isr = new InputStreamReader(System.in) ;
```

το `System.in` που είναι `byte stream` μετατρέπεται σε `stream` χαρακτήρων.

Με την εντολή :

```
BufferedReader br = new BufferedReader(isr) ;
```

το αντικείμενο τύπου `BufferedReader` `br` είναι πλέον ένα `stream` χαρακτήρων, το οποίο συνδέεται με το πληκτρολόγιο, μέσω του `System.in`.

Το ίδιο αποτέλεσμα έχουμε με ΜΙΑ εντολή :

```
BufferedReader br = new BufferedReader(new  
InputStreamReader(System.in)) ;
```

Το αντικείμενο τύπου `BufferedReader` συνδέεται και με τις αντίστοιχες μεθόδους. Έτσι, η μορφή που μπορεί να πάρει η κλήση της μεθόδου για διάβασμα χαρακτήρων σε μορφή χαρακτήρων είναι :

`br.read()` για το διάβασμα ενός χαρακτήρα

`br.readLine()` για το διάβασμα μιας Γραμμής Χαρακτήρων (Συμβολοσειράς).

ενώ για την εμφάνιση των χαρακτήρων ή των Συμβολοσειρών χρησιμοποιούνται οι μέθοδοι `print()` και `println()`.

Παρατήρηση

Στα επαγγελματικά προγράμματα η εμφάνιση γίνεται με τη δημιουργία ενός αντικειμένου τύπου `PrintWriter` και τη χρήση των μεθόδων `print()` και `println()`.

Παράδειγμα 1

Με το παρακάτω πρόγραμμα διαβάζουμε χαρακτήρες με τη μέθοδο `read()` μέχρι να δώσουμε τελεία και τους εμφανίζουμε.

Πρόγραμμα

```
package readchars;
import java.io.*;
public class ReadChars {

    public static void main(String[] args)
        throws java.io.IOException{
        char ch;

        System.out.print("Enter Your Name, Finish With . : ");
        BufferedReader br = new BufferedReader(new
            InputStreamReader(System.in));

        do {
            ch = (char)br.read();
            System.out.print(ch);
        }
        while ( ch != '.');
        System.out.println();
    }
}
```

Έξοδος Προγράμματος

```
run:
Enter Your Name, Finish With . : Kostas Goulianas.
Kostas Goulianas.
BUILD SUCCESSFUL (total time: 9 seconds)
```

Παράδειγμα 2

Με το επόμενο πρόγραμμα δηλώνουμε μια μεταβλητή `line` τύπου `String`, στην οποία αποθηκεύουμε τη συμβολοσειρά που διαβάζουμε με τη μέθοδο `readLine()`.

Πρόγραμμα

```
package readstrings;
import java.io.BufferedReader;
import java.io.InputStreamReader;

public class ReadStrings {

    public static void main(String[] args)
        throws java.io.IOException{
        String line;

        System.out.println("Enter Lines, Finish With stop : ");
        BufferedReader br = new BufferedReader(new
            InputStreamReader(System.in));

        do {
            line = br.readLine();
        }
        while ( !line.equals("stop") );
        // System.out.println();
    }
}
```

Έξοδος Προγράμματος

```
run:
Enter Lines, Finish With stop :
line 1
line 2
line 3
stop
BUILD SUCCESSFUL (total time: 18 seconds)
```

Παρατήρηση 2

Και στα δύο παραδείγματα θα πρέπει να εισάγουμε τις κλάσεις **BufferedReader** και **InputStreamReader** με τις εντολές :

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
```

ή με την εντολή :

```
import java.io.*;
```

10.3 Μετατροπή Συμβολοσειρών σε Απλούς Τύπους - Wrappers

Η μέθοδος `println()` στην εντολή `System.out.println()` μετατρέπει αυτόματα το περιεχόμενο μιας μεταβλητής σε αναγνώσιμη μορφή. Δεν μπορεί να γίνει όμως το ίδιο και με τις συμβολοσειρές που διαβάζονται με τη μέθοδο `readLine()`, ώστε να μετατραπούν σε δεδομένα απλών τύπων (παράδειγμα η συμβολοσειρά 123 να μετατραπεί στον ακέραιο αριθμό 123), γιατί οι απλοί τύποι δεν είναι αντικείμενα. Χρειάζονται λοιπόν οι αντίστοιχοι **μετατροπείς** σε τύπους `double`, `float`, `long`, `int`, `short`, `byte`, `char` και `boolean`. Για το σκοπό αυτό χρησιμοποιούνται οι αντίστοιχες κλάσεις για κάθε απλό τύπο με τις αντίστοιχες μεθόδους, οι οποίες είναι οι παρακάτω :

Μετατροπέας - Wrapper

```
Double.parseDouble(<string>)
Float.parseFloat(<string>)
Long.parseLong(<string>)
Integer.parseInt(<string>)
Short.parseShort(<string>)
Byte.parseByte(<string>)
```

Αποτέλεσμα

```
Μετατροπή του <string> σε double
Μετατροπή του <string> σε float
Μετατροπή του <string> σε long
Μετατροπή του <string> σε int
Μετατροπή του <string> σε short
Μετατροπή του <string> σε byte
```

10.3.1 Πρόγραμμα με Μεθόδους Μετατροπής String σε Δεδομένα Απλών Τύπων

Να γίνει πρόγραμμα που θα ζητάει απ' το χρήστη να εισάγει έναν ακέραιο αριθμό από το πληκτρολόγιο, θα μετατρέπει τη συμβολοσειρά σε ακέραιο και θα εμφανίζει την τιμή του.

Πρόγραμμα

```
package readinteger;
import java.io.*;
public class ReadInteger {
/* Πρόγραμμα που διαβάζει μια συμβολοσειρά απ' το πληκτρολόγιο, τη μετατρέπει
σε ακέραιο αριθμό, τον οποίο και εμφανίζει
*/
public static void main(String[] args)
                throws java.io.IOException{
    String line;
    int number;
    System.out.print("Δώσε έναν ακέραιο αριθμό : ");
    // Δημιουργία Αντικειμένου για Μετατροπή Εισόδου σε stream Χαρακτήρων
    BufferedReader br = new BufferedReader(new
                InputStreamReader(System.in));
    // Εισαγωγή Συμβολοσειράς
    line = br.readLine();
    // Μετατροπή string Εισόδου σε Ακέραιο
    number = Integer.parseInt( line );
    // Εμφάνιση Τιμής Ακεραίου Μετά τη Μετατροπή
    System.out.println("Ο αριθμός που έδωσες είναι : " + number);
}
}
```

Έξοδος Προγράμματος

```
run:
Δώσε έναν ακέραιο αριθμό : 12345
Ο αριθμός που έδωσες είναι : 12345
BUILD SUCCESSFUL (total time: 4 seconds)
```

```
run:
Δώσε έναν ακέραιο αριθμό : 123.45
Exception in thread "main" java.lang.NumberFormatException: For input string:
"123.45"
    at
    java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
    at java.lang.Integer.parseInt(Integer.java:580)
    at java.lang.Integer.parseInt(Integer.java:615)
    at readinteger.ReadInteger.main(ReadInteger.java:18)
Java Result: 1
BUILD SUCCESSFUL (total time: 4 seconds)
```

```
run:
Δώσε έναν ακέραιο αριθμό : kostas
Exception in thread "main" java.lang.NumberFormatException: For input string:
"kostas"
    at
    java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
    at java.lang.Integer.parseInt(Integer.java:580)
    at java.lang.Integer.parseInt(Integer.java:615)
    at readinteger.ReadInteger.main(ReadInteger.java:18)
Java Result: 1
BUILD SUCCESSFUL (total time: 4 seconds)
```

Παρατήρηση

Η εισαγωγή πραγματικού αριθμού και συμβολοσειράς προκάλεσε ένα σφάλμα `NumberFormatException`, το οποίο πρέπει να ελεγχθεί, όπως φαίνεται στο επόμενο παράδειγμα :

10.3.2 Πρόγραμμα με Μεθόδους Μετατροπής String σε Δεδομένα Απλών Τύπων και Έλεγχο για Πιθανή Εξαίρεση

Να τροποποιηθεί το προηγούμενο πρόγραμμα που ζητάει απ' το χρήστη να εισάγει έναν ακέραιο αριθμό από το πληκτρολόγιο, μετατρέπει τη συμβολοσειρά σε ακέραιο και εμφανίζει την τιμή του, έτσι ώστε **να κάνει και τον έλεγχο για εξαιρέσεις**.

Πρόγραμμα

```
package readintegertrycatch;
import java.io.*;
public class ReadIntegerTryCatch {
    /* Πρόγραμμα που διαβάζει μια συμβολοσειρά απ' το πληκτρολόγιο, τη μετατρέπει
    σε ακέραιο αριθμό, τον οποίο και εμφανίζει κάνοντας και τον έλεγχο για πιθανή
    εξαίρεση */
    public static void main(String[] args) {
        String line;
        int number;
        System.out.print("Δώσε έναν ακέραιο αριθμό : ");
        // Δημιουργία Αντικειμένου για Μετατροπή Εισόδου σε stream Χαρακτήρων
        BufferedReader br = new BufferedReader(new
            InputStreamReader(System.in));

        try {
            // Εισαγωγή Συμβολοσειράς
            line = br.readLine();
            // Μετατροπή string Εισόδου σε Ακέραιο
            number = Integer.parseInt( line );
            // Εμφάνιση Τιμής Ακεραίου Μετά τη Μετατροπή
            System.out.println("Ο αριθμός που έδωσες είναι : " + number);
        }
        catch ( Exception obj){
            System.out.println("Σφάλμα Εισόδου - Εξόδου");
        }
    }
}
```

Έξοδος Προγράμματος

```
run:
Δώσε έναν ακέραιο αριθμό : 12345
Ο αριθμός που έδωσες είναι : 12345
BUILD SUCCESSFUL (total time: 4 seconds)
```

```
run:
Δώσε έναν ακέραιο αριθμό : 123.45
Σφάλμα Εισόδου - Εξόδου
BUILD SUCCESSFUL (total time: 6 seconds)
```

```
run:
Δώσε έναν ακέραιο αριθμό : kostas
Σφάλμα Εισόδου - Εξόδου
BUILD SUCCESSFUL (total time: 6 seconds)
```

Παρατήρηση

Η εισαγωγή του αριθμού και ο έλεγχος για πιθανό σφάλμα θα μπορούσαν να ανήκουν σε μια `static` μέθοδο, όπως φαίνεται στο επόμενο παράδειγμα :

10.3.3 Πρόγραμμα με `static` Μέθοδο για Διάβασμα Ακεραίου

Να τροποποιηθεί το προηγούμενο πρόγραμμα που ζητάει απ' το χρήστη να εισάγει έναν ακέραιο αριθμό από το πληκτρολόγιο, μετατρέπει τη συμβολοσειρά σε ακέραιο και εμφανίζει την τιμή του, κάνοντας και τον έλεγχο για εξαιρέσεις, ώστε το διάβασμα και η μετατροπή της συμβολοσειράς σε ακέραιο να γίνεται με μια μέθοδο. Η `main()` απλώς θα εμφανίζει το μήνυμα εισαγωγής του αριθμού και θα εμφανίζει την τιμή του .

Πρόγραμμα

```
package readintegertrycatchmethod;
import java.io.*;
public class ReadIntegerTryCatchMethod {
    /*
    Πρόγραμμα που διαβάζει μια συμβολοσειρά απ' το πληκτρολόγιο, τη μετατρέπει
    σε ακέραιο αριθμό, τον οποίο και εμφανίζει κάνοντας και τον έλεγχο για πιθανή
    εξαίρεση με την κλήση της μεθόδου readInteger()
    */

    static int readInteger() {
        String line;
        // Δημιουργία Αντικειμένου για Μετατροπή Εισόδου σε stream Χαρακτήρων
        BufferedReader br = new BufferedReader(new
            InputStreamReader(System.in));
        try {
            // Εισαγωγή Συμβολοσειράς
            line = br.readLine();
            // Μετατροπή string Εισόδου σε Ακέραιο
            int i = Integer.parseInt( line );
            return i;
        }
        catch ( Exception obj){
            System.out.println("Σφάλμα Εισόδου - Εξόδου");
            return -1;
        }
    }

    public static void main(String[] args) {

        int number;

        System.out.print("Δώσε έναν ακέραιο αριθμό : ");

        // Κλήση της μεθόδου readInteger()
        number = readInteger();

        // Εμφάνιση Τιμής Ακεραίου Μετά τη Μετατροπή
        System.out.println("Ο αριθμός που έδωσες είναι : " + number);
    }
}
```


Έξοδος Προγράμματος

```
run:
Δώσε έναν ακέραιο αριθμό : 12345
Ο αριθμός που έδωσες είναι : 12345
BUILD SUCCESSFUL (total time: 4 seconds)
```

```
run:
Δώσε έναν ακέραιο αριθμό : 123.45
Σφάλμα Εισόδου - Εξόδου
Ο αριθμός που έδωσες είναι : -1
BUILD SUCCESSFUL (total time: 5 seconds)
```

```
run:
Δώσε έναν ακέραιο αριθμό : kostas
Σφάλμα Εισόδου - Εξόδου
Ο αριθμός που έδωσες είναι : -1
BUILD SUCCESSFUL (total time: 4 seconds)
```

10.3.4 Κλάση με static Μεθόδους Μετατροπής String σε Δεδομένα Απλών Τύπων και Έλεγχο για Πιθανή Εξαίρεση

Να γίνει μια κλάση, η οποία θα περιέχει μεθόδους για εισαγωγή δεδομένων οποιουδήποτε απλού τύπου από το πληκτρολόγιο. Να χρησιμοποιηθούν οι μετατροπείς και να γίνει έλεγχος για εξαιρέσεις.

Κλάση userInput

```
import java.io.*;

class userInput { //Class gia eisagogi dedomenwn apo to pliktrologio
    static String getString() { //Methodos gia eisagogi String
        String line;
        InputStreamReader input = new InputStreamReader(System.in);
        BufferedReader in = new BufferedReader(input);
        try
        {
            line = in.readLine();
            return line;
        }
        catch(Exception e)
        {
            return "Exception";
        }
    }

    static byte getByte() { //Methodos gia eisagogi short
        String line;
        InputStreamReader input = new InputStreamReader(System.in);
        BufferedReader in = new BufferedReader(input);
        try{
            line = in.readLine();
            byte b = Byte.parseByte(line);
            return b;
        }
        catch(Exception e){
            return -1;
        }
    }
}
```

```

static short getShort() { //Methodos gia eisagogi short
    String line;
    InputStreamReader input = new InputStreamReader(System.in);
    BufferedReader in = new BufferedReader(input);
    try{
        line = in.readLine();
        short s = Short.parseShort(line);
        return s;
    }
    catch(Exception e){
        return -1;
    }
}

```

```

static int getInteger() { //Methodos gia eisagogi Integer
    String line;
    InputStreamReader input = new InputStreamReader(System.in);
    BufferedReader in = new BufferedReader(input);
    try{
        line = in.readLine();
        int i = Integer.parseInt(line);
        return i;
    }
    catch(Exception e){
        return -1;
    }
}

```

```

static short getLong() { //Methodos gia eisagogi long
    String line;
    InputStreamReader input = new InputStreamReader(System.in);
    BufferedReader in = new BufferedReader(input);
    try{
        line = in.readLine();
        long l = Long.parseLong(line);
        return l;
    }
    catch(Exception e){
        return -1;
    }
}

```

```

static float getFloat() { //Methodos gia eisagogi float
    String line;
    InputStreamReader input = new InputStreamReader(System.in);
    BufferedReader in = new BufferedReader(input);
    try{
        line = in.readLine();
        float f = Float.parseFloat(line);
        return f;
    }
    catch(Exception e){
        return -1;
    }
}

```

```

static double getDouble() { //Methodos gia eisagogi double
    String line;
    InputStreamReader input = new InputStreamReader(System.in);
    BufferedReader in = new BufferedReader(input);
    try{
        line = in.readLine();
        double d = Double.parseDouble(line);
        return d;
    }
    catch(Exception e){
        return -1;
    }
}

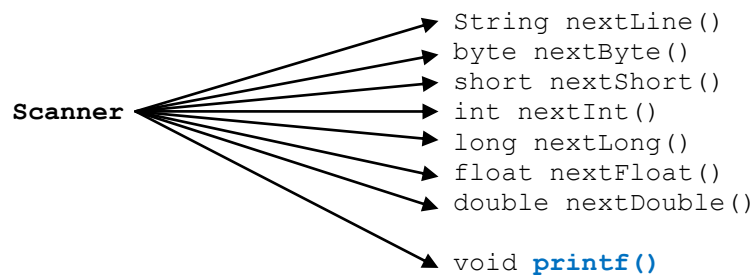
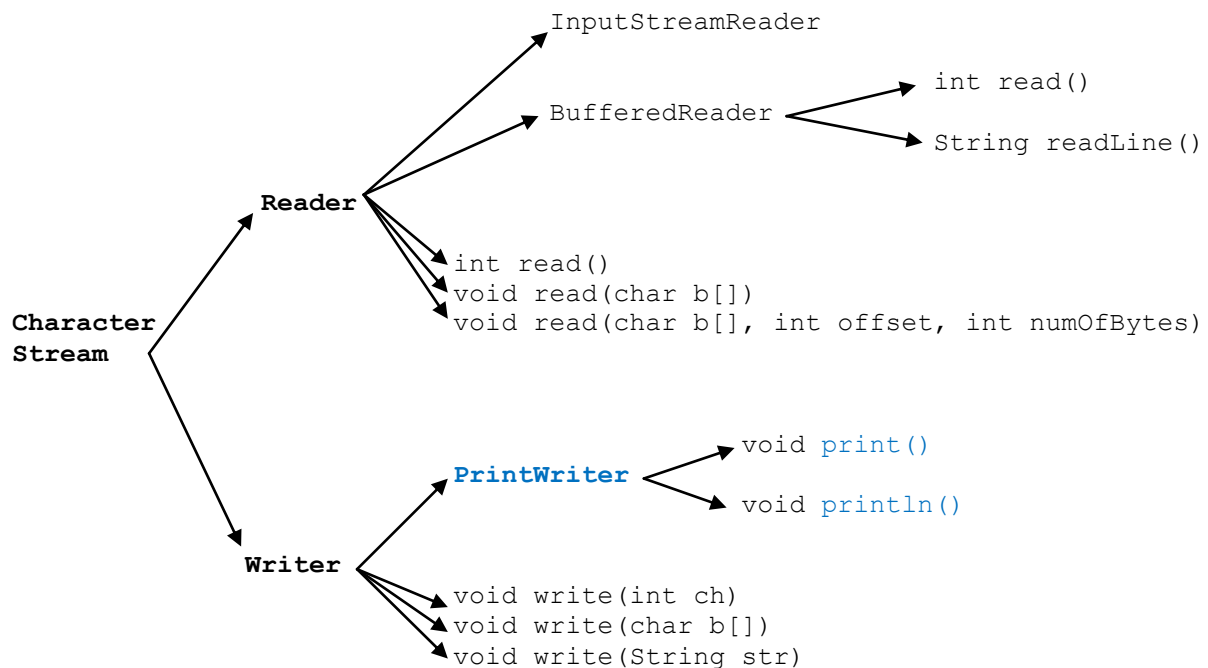
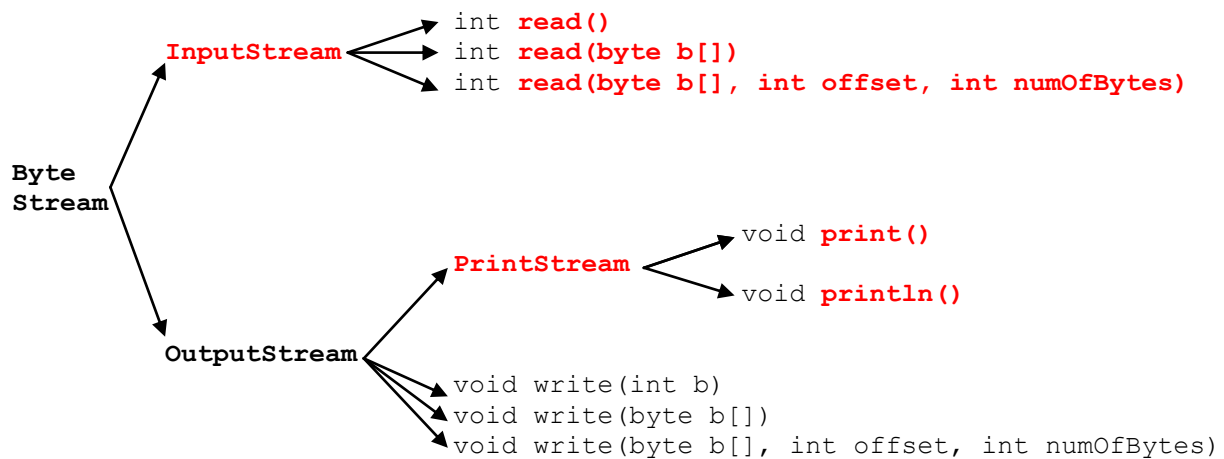
```

```

static boolean getBoolean() { //Methodos gia eisagogi String
    String line;
    InputStreamReader input = new InputStreamReader(System.in);
    BufferedReader in = new BufferedReader(input);
    try
    {
        line = in.readLine();
        return (Boolean.valueOf(line)).booleanValue();
    }
    catch(Exception e)
    {
        return false;
    }
}

```

10.4 Σχεδιάγραμμα Κλάσεων, Υποκλάσεων και Μεθόδων για τις κλάσεις Byte Streams, Streams Χαρακτήρων και Scanner



System.in - Είσοδος - Πληκτρολόγιο - Αντικείμενο Τύπου **InputStream**
System.out - Έξοδος - Οθόνη - Αντικείμενο Τύπου **PrintStream**

10.5 Είσοδος – Έξοδος Δεδομένων με τη Χρήση της Κλάσης Scanner

Η κλάση **Scanner** παρέχει έναν πιο απλό τρόπο για είσοδο δεδομένων απ' το πληκτρολόγιο και έναν πιο ελεγχόμενο τρόπο εμφάνισης του περιεχομένου κάποιων μεταβλητών. Για να τη χρησιμοποιήσουμε, θα πρέπει πρώτα να την εισάγουμε με την εντολή

```
import java.util.Scanner;
```

Μετά, μπορούμε να συνδέσουμε με το πληκτρολόγιο, μέσω του `System.in`, ένα αντικείμενο τύπου **Scanner** με την εντολή

```
Scanner <object> = new Scanner(System.in);
```

Το αντικείμενο `<object>` τύπου `Scanner` συνδέεται και με τις αντίστοιχες μεθόδους. Οι μέθοδοι που διαθέτει η κλάση `Scanner` για εισαγωγή δεδομένων είναι :

Μέθοδος	Αποτέλεσμα
<code>nextLine()</code>	Διάβασμα Συμβολοσειράς
<code>nextByte()</code>	Διάβασμα Αριθμού, Αποθήκευση σε Μεταβλητή <code>byte</code>
<code>nextShort()</code>	Διάβασμα Αριθμού, Αποθήκευση σε Μεταβλητή <code>short</code>
<code>nextInt()</code>	Διάβασμα Αριθμού, Αποθήκευση σε Μεταβλητή <code>int</code>
<code>nextLong()</code>	Διάβασμα Αριθμού, Αποθήκευση σε Μεταβλητή <code>long</code>
<code>nextFloat()</code>	Διάβασμα Αριθμού, Αποθήκευση σε Μεταβλητή <code>float</code>
<code>nextDouble()</code>	Διάβασμα Αριθμού, Αποθήκευση σε Μεταβλητή <code>double</code>

Η μέθοδος για την εμφάνιση των δεδομένων (εκτός των μεθόδων `print()` και `println()`) είναι η μέθοδος `printf()`, η οποία, σε αντίθεση με τις 2 προηγούμενες μεθόδους εμφάνισης, δίνει τη δυνατότητα στο χρήστη να επιλέξει τον **τρόπο εμφάνισης** των δεδομένων που είναι αποθηκευμένα σε κάποιες μεταβλητές, των οποίων το περιεχόμενο θέλει να εμφανίσει. Η σύνταξή της διαφέρει από τη μέθοδο `println()`. Αντί να υπάρχουν **μηνύματα** και **ονόματα μεταβλητών** συνδεδεμένα με το '+', όπως στην εντολή `System.out.println()`, στην εντολή `System.out.printf()` υπάρχει μια συμβολοσειρά με **μηνύματα** και **κωδικούς εμφάνισης** των τιμών των μεταβλητών και ακολουθεί η **λίστα των μεταβλητών**, τις τιμές των οποίων θέλουμε να εμφανίσουμε, **χωρισμένες με κόμμα**. Ο κωδικός εμφάνισης των τιμών μεταβλητών τύπου `byte`, `short`, `int` και `long` είναι `%d`, ενώ ο κωδικός εμφάνισης των τιμών μεταβλητών τύπου `float` και `double` είναι `%f`. Βάζοντας έναν ακέραιο αριθμό μεταξύ του `%` και του `d`, στον κωδικό `%d`, ο ακέραιος αριθμός που περιέχει η μεταβλητή που αντιστοιχεί σ' αυτό τον κωδικό θα καταλάβει τόσες θέσεις στην οθόνη, όσες και ο αριθμός στον κωδικό `%d`. Αν ο ακέραιος αριθμός που περιέχει η μεταβλητή δεν έχει τόσα ψηφία, οι αριστερές θέσεις θα γεμίσουν με τα αντίστοιχα κενά. Π.χ. για τον αριθμό 123 με κωδικό `%5d`, όπου ο αριθμός 5 στον κωδικό `%5d` = <αριθμός_ψηφίων_στην_οθόνη> είναι μεγαλύτερος του 3, θα εμφανιστεί το `^123`, όπου `^` = κενό. Αν ο <αριθμός_ψηφίων_στην_οθόνη> είναι μικρότερος των ψηφίων του αριθμού, θα εμφανιστεί ο αριθμός χωρίς κενά. Αντίστοιχα, στον κωδικό `%f` μεταξύ του `%` και του `f`, μπορούν να υπάρχουν 2 αριθμοί χωρισμένοι με τελεία, όπου ο πρώτος δηλώνει τις **συνολικές θέσεις** που θα καταλάβει ο αριθμός κινητής υποδιαστολής στην οθόνη, ενώ ο δεύτερος τον **αριθμό των δεκαδικών ψηφίων**. Π.χ. ο κωδικός `%30.20f` σημαίνει πως ο αριθμός που περιέχει η μεταβλητή που αντιστοιχεί σ' αυτό τον κωδικό θα καταλάβει 30

θέσεις συνολικά στην οθόνη, απ' τις οποίες οι 20 αντιστοιχούν σε δεκαδικά ψηφία. Αν τα μη μηδενικά δεκαδικά ψηφία του αριθμού είναι λιγότερα από όσα έχουμε προσδιορίσει, οι υπόλοιπες θέσεις γεμίζουν με μηδενικά, ενώ, αν είναι λιγότερα τα ψηφία πριν την υποδιαστολή, οι υπόλοιπες θέσεις γεμίζουν με κενά.

10.5.1 Παράδειγμα Εισόδου – Εξόδου Δεδομένων με τη Χρήση της Κλάσης Scanner

Να γραφεί πρόγραμμα που διαβάζει δεδομένα τύπου `byte`, `short`, `int`, `long`, `float`, `double` και `String` με τη χρήση της κλάσης `Scanner` και τα εμφανίζει με τη χρήση της μεθόδου `printf()`.

Πρόγραμμα

```
package scannerclass;
import java.util.Scanner;
public class ScannerClass {
/* Πρόγραμμα που εισάγει διάφορα δεδομένα με τη χρήση της κλάσης Scanner
και τα εμφανίζει με τη χρήση της μεθόδου printf() */
    public static void main(String[] args) {
        // Δημιουργία αντικειμένου τύπου Scanner
        Scanner ob = new Scanner(System.in);

        // Εισαγωγή Δεδομένων διαφόρων τύπων
        System.out.print("Δώσε ένα Όνομα - Μέθοδος nextLine() : ");
        String name1 = ob.nextLine();
        System.out.print("Δώσε έναν αριθμό για μεταβλητή byte : ");
        byte b = ob.nextByte();
        System.out.print("Δώσε έναν αριθμό για μεταβλητή short : ");
        short s=ob.nextShort();
        System.out.print("Δώσε έναν αριθμό για μεταβλητή int : ");
        int i=ob.nextInt();
        System.out.print("Δώσε έναν αριθμό για μεταβλητή long : ");
        long l=ob.nextLong();
        System.out.print("Δώσε έναν αριθμό για μεταβλητή float : ");
        float f = ob.nextFloat();
        System.out.print("Δώσε έναν αριθμό για μεταβλητή double : ");
        double d = ob.nextDouble();

        // Εμφάνιση τιμής μεταβλητής String με τη χρήση της μεθόδου println()
        System.out.println("\nΤο Όνομα με τη nextLine() είναι : " + name1 );

        // Εμφάνιση τιμών υπολοίπων μεταβλητών με τη χρήση της μεθόδου printf()
        System.out.printf("nΕμφάνιση τιμών υπολοίπων μεταβλητών με τη χρήση της
μεθόδου printf()\n");
        System.out.printf("Η τιμή της byte Μεταβλητής με format 5d είναι :
%5d\n", b);
        System.out.printf("Η τιμή της short Μεταβλητής με format 10d είναι :
%10d\n", s);
        System.out.printf("Η τιμή της int Μεταβλητής με format 15d είναι :
%15d\n", i);
        System.out.printf("Η τιμή της long Μεταβλητής με format 20d είναι :
%20d\n", l);
        System.out.printf("Η τιμή της float Μεταβλητής με format 18.16f είναι :
%18.16f\n", f);
        System.out.printf("Η τιμή της double Μεταβλητής με format 30.20f είναι :
%30.20f\n", d);
    }
}
```

Έξοδος Προγράμματος

```
run:
Δώσε ένα Όνομα - Μέθοδος nextLine() : Georgios Georgiou
Δώσε έναν αριθμό για μεταβλητή byte : 12
Δώσε έναν αριθμό για μεταβλητή short : 123
Δώσε έναν αριθμό για μεταβλητή int : 1234
Δώσε έναν αριθμό για μεταβλητή long : 123456789
Δώσε έναν αριθμό για μεταβλητή float : 123,45
Δώσε έναν αριθμό για μεταβλητή double : 123456789,12345
```

Το Όνομα με τη nextLine() είναι :Georgios Georgiou

```
Εμφάνιση τιμών υπολοίπων μεταβλητών με τη χρήση της μεθόδου printf()
Η τιμή της byte Μεταβλητής με format 5d είναι : 12
Η τιμή της short Μεταβλητής με format 10d είναι : 123
Η τιμή της int Μεταβλητής με format 15d είναι : 1234
Η τιμή της long Μεταβλητής με format 20d είναι : 123456789
Η τιμή της float Μεταβλητής με format 18.16f είναι : 123.4499969482421900
Η τιμή της double Μεταβλητής με format 30.20f είναι :
123456789.123450000000000000000000
BUILD SUCCESSFUL (total time: 44 seconds)
```

10.5.2 Παράδειγμα Δημιουργίας Μεθόδου Εισόδου Δεδομένων τύπου int

Να γραφεί πρόγραμμα που να καλεί τη μέθοδο `scanInteger()` της κλάσης `ScannerUserInput`, η οποία **διαβάζει** έναν αριθμό τύπου `int` με τη χρήση της κλάσης `Scanner` και τον **εμφανίζει** με τη χρήση της μεθόδου `printf()`.

Κλάση `ScannerInput` - `main()`

```
package scannerinput;
public class ScannerInput {
    public static void main(String[] args) {
        System.out.print("Δώσε έναν ακέραιο αριθμό : ");
        int i = ScannerUserInput.scanInteger();
        System.out.printf("Ο ακέραιος αριθμός είναι : : %15d\n", i);
    }
}
```

Κλάση `ScannerUserInput` - Μέθοδος `scanInteger()`

```
package scannerinput;
import java.util.Scanner;
public class ScannerUserInput {

    static int scanInteger(){
        // Δημιουργία αντικειμένου τύπου Scanner
        Scanner ob = new Scanner(System.in);
        try{
            int i = ob.nextInt();
            return i;
        }
        catch(Exception e){
            return -1;
        }
    }
}
```

Έξοδος Προγράμματος

```
run:
Δώσε έναν ακέραιο αριθμό : 12345
Ο ακέραιος αριθμός είναι : : 12345
BUILD SUCCESSFUL (total time: 5 seconds)
```

```
run:
Δώσε έναν ακέραιο αριθμό : 5.0
Ο ακέραιος αριθμός είναι : : -1
BUILD SUCCESSFUL (total time: 3 seconds)
```

```
run:
Δώσε έναν ακέραιο αριθμό : abc
Ο ακέραιος αριθμός είναι : : -1
BUILD SUCCESSFUL (total time: 6 seconds)
```

ΑΣΚΗΣΗ 10.1

Με τη χρήση της **Κλάσης Scanner**, να συμπληρωθεί η κλάση **ScannerUserInput** με τις υπόλοιπες μεθόδους για εισαγωγή δεδομένων οποιουδήποτε απλού τύπου από το πληκτρολόγιο κάνοντας και τον έλεγχο για εξαιρέσεις.