

Solutions

Solution 1.

```
import java.io.*;

public class Lecture {

    private static String lireLigne (BufferedReader f) {
        String resultat = new String();

        try {
            resultat = f.readLine();
        }

        catch(Exception e) {
            System.err.println("Erreur de lecture");
            System.exit(1);
        }

        return resultat;
    }

    private static int lireEntier (BufferedReader f) {
        return Integer.parseInt(lireLigne(f));
    }

    private static String majuscule(BufferedReader f) {
        return (lireLigne(f)).toUpperCase();
    }

    private static int somme(BufferedReader f) {
        int resultat = 0;
        String s = lireLigne(f);
        while(s.length() != 0) {
            resultat += Integer.parseInt(s);
            s = lireLigne(f);
        }
        return resultat;
    }

    public static void main(String[] args) {
        BufferedReader flot = new BufferedReader(new InputStreamReader(System.in));

        // System.out.println("Entrer une phrase :");
        // System.out.println("--> " + majuscule(flott));
    }
}
```

```
System.out.println("Entrer des nombres.");
System.out.println("-->" + somme(flott));
```

```
}
```

```
}
```

Solution 2.

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

```
public class LectureFichier {
```

```
/**
```

```
 * traitement sommaire des erreurs : affichage et sortie
```

```
 */
```

```
private static void erreur (String s) {
```

```
    System.err.println (s);
```

```
    System.exit (1);
```

```
}
```

```
/**
```

```
 * traitement d'une ligne : ici conversion en majuscule
```

```
 */
```

```
private static void traiterLigne (String l) {
```

```
    System.out.println (l.toUpperCase ());
```

```
}
```

```
/**
```

```
 * traitement de l'entree : lecture ligne par ligne
```

```
 * et traitement de chaque fin de ligne lue.
```

```
 */
```

```
private static void traiterLecture (BufferedReader entree) {
```

```
    for (;;) {
```

```
        try {
```

```
            String ligne = entree.readLine ();
```

```
            if (ligne == null)
```

```
                break;
```

```
            traiterLigne (ligne);
```

```
        }
```

```
        catch (Exception e) {
```

```
            erreur ("Erreur de lecture");
```

```
        }
```

```
    }
```

```
}
```

```

private static int compterLignes (BufferedReader entree) {
    int cpt = 0;

    for (;;) {
        try {
            String ligne = entree.readLine ();

            if (ligne == null)
                break;
            else
                cpt++;
        }
        catch (Exception e) {
            erreur ("Erreur_de_lecture");
        }
    }
    return cpt;
}

/**
 * Creation d un lecteur avec tampon associee a un lecteur sur fichier .
 * Ce dernier est initialise a partir d'un nom de fichier
 */
private static BufferedReader ouvrirFichier (String nom) {
    BufferedReader entree = null;
    try {
        entree = new BufferedReader (new FileReader (nom));
    }
    catch (Exception e) {
        erreur ("Erreur_Ouverture_de_fichier_" + nom);
    }
    return entree;
}

private static void fermerEntree (BufferedReader entree) {
    try {
        entree.close ();
    }
    catch (Exception e) {
        erreur ("erreur_de_fermeture_de_fichier");
    }
}

/**

```

* Exemple de traitement; le nom de fichier est le premier argument
* de la commande.

*/

```
public static void main (String args[]) {  
    if (args.length != 1)  
        erreur ("Usage: <nom_de_fichier>");  
    BufferedReader entree = ouvrirFichier (args [0]);  
    System.out.println("Le_fichier " + args[0] + ".a"  
        + compterLignes(entree));  
    fermerEntree (entree);  
}  
}
```

Solution 3.

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

```
public class EcritureFichier {
```

```
    private static void erreur(String s) {  
        System.err.println(s);  
        System.exit(1);  
    }
```

```
    private static PrintWriter ouvrirFichier(String nom) {  
        PrintWriter sortie = null;  
        try {  
            sortie = new PrintWriter(new BufferedWriter( new FileWriter(nom)));  
        }  
        catch(Exception e) {  
            erreur("ouverture_de_fichier" + nom);  
        }  
        return sortie;  
    }
```

```
    private static void fermerSortie(PrintWriter sortie) {  
        try {  
            sortie.close ();  
        }  
        catch(Exception e) {  
            erreur("erreur_fermeture_lecteur");  
        }  
    }
```

```
    private static String lireLigne (BufferedReader f) {  
        String resultat = new String();
```

```

    try {
        resultat = f.readLine();
    }

    catch(Exception e) {
        System.err.println("Erreur de lecture");
        System.exit(1);
    }

    return resultat;
}

private static void ecrire(PrintWriter sortie , BufferedReader f) {

    for (;;) {
        try {
            String ligne = lireLigne(f);

            if (ligne == null)
                break;
            else
                sortie . println (ligne );
        }
        catch (Exception e) {
            erreur ("Erreur de lecture");
        }
    }
}

public static void main(String args[]) {
    if(args.length != 1)
        erreur("Usage: ./<fichier>");
    PrintWriter sortie = ouvrirFichier(args [0]);
    BufferedReader flot = new BufferedReader(new InputStreamReader(System.in));
    ecrire ( sortie , flot );
    fermerSortie( sortie );
}
}

```