

Algorithmique & Programmation

Extensions iJava : fichiers textuels

yann.secq@univ-lille.fr

ABIDI Sofiene, ALMEIDA COCO Amadeu, BONEVA Iovka, CASTILLON Antoine,
DELECROIX Fabien, LEPRETRE Éric, Timothé ROUZÉ, SANTANA MAIA Deise,
SECQ Yann

Fonctions prédéfinies

Sorties (terminal)

```
void print(String s)
void print(int i)
void print(long l)
void print(float f)
void print(double d)
void print(char c)
void print(boolean b)
```

```
void println()
void println(String s)
void println(int i)
void println(long l)
void println(float f)
void println(double d)
void println(char c)
void println(boolean b)
```

Entrées (clavier)

```
int readInt()
long readLong()
float readFloat()
double readDouble()
String readString()
char readChar()
```

Sur les chaînes de caractères

```
boolean equals(String s1, String s2)
int compare(String s1, String s2)
int length(String s)
String substring(String s, int start, int end)
char charAt(String s, int idx)
String toUpperCase(String s)
String toLowerCase(String s)
```

Sur les nombres

```
double random()
int stringToInt(String s)
int charToInt(char c)
```

Sur les tableaux

```
int length(<tableau> t)
int length(<tableau> t,
            int dim)
```

Extensions iJava

- Comment utiliser d'autres fonctions prédéfinies, mais non disponibles par défaut ?
- Notion de bibliothèque (ou librairie, de l'anglais *library*)
- Ensemble de type et fonctions définies dans un autre espace de nommage ...
- Pour iJava, l'espace `extensions`
- Exemple d'usage avec la manipulation de fichiers

Extensions iJava

- Notion de fichier (cf. cours de système !)
- Gérer des fichiers textes
- Gérer des fichiers CSV
- Usages potentiels dans le cadre des projets

Notion de fichier

- Un ensemble de données ...
- Concept polymorphe avec UNIX !
- Limitons-nous aux fichiers textuels
- Stockage pérenne d'information textuelles
- Opérations : lecture et écriture

Notion de fichiers

- Deux approches possibles avec `iJava`
 - Lecture de chaînes de caractères ligne par ligne
 - Lecture sous forme de grille pour les fichiers de type CSV

Ligne par ligne

- Pas un type de base, nécessaire de l'**importer** explicitement !
- Utiliser le type `File`
- *Tant que « prêt », lire la prochaine ligne*
- Doc. technique ... pas terrible :(

`extensions.File`

`newFile(String filename)`

`String`

`readLine(extensions.File file)`

`boolean`

`ready(extensions.File file)`

Réécrivons cat !

!! =>

```
import extensions.File;
class Cat extends Program {
    void algorithm() {
        final String FILENAME = "Cat.java";
        File f = new File(FILENAME);
        int nbLines = 0, nbChars = 0;
        while (ready(f)) {
            String currentLine = readLine(f);
            nbLines++;
            nbChars = nbChars + length(currentLine);
            println(currentLine);
        }
        println("-----");
        println("Le fichier "+FILENAME+" contient "+
            nbLines+" lignes et "+nbChars+" caractères.");
    }
}
```


Exécution de Cat

```
yannsecq@YANNs-MacBook-Pro fouillis % java -cp ../ap-last-version.jar:. Cat
import extensions.File;
class Cat extends Program {
    void algorithm() {
        final String FILENAME = "Cat.java";
        File f = new File(FILENAME);
        int nbLines = 0, nbChars = 0;
        while (ready(f)) {
            String currentLine = readLine(f);
            nbLines++;
            nbChars = nbChars + length(currentLine);
            println(currentLine);
        }
        println("-----");
        println("Le fichier "+FILENAME+" contient "+
            nbLines+" lignes et "+nbChars+" caractères.");
    }
}

-----
Le fichier Cat.java contient 17 lignes et 483 caractères.
```

Fichiers structurés : CSV

- Type de fichier permettant de représenter une base de données très simplifiée (une table)
- Fichier texte constitué de lignes dont les éléments sont séparés par des (points) virgules
- Utile pour sauvegarder une information structurée de manière pérenne (ie. entre différentes exécutions)

Type fichier CSV

- Utiliser le type `CSVFile`
- S'apparente à un `String[][]` en mémoire
- Documentation technique ... pas terrible :(

<code>int</code>	<code>columnCount(CSVFile table)</code>
<code>int</code>	<code>columnCount(CSVFile table, int idxLine)</code>
<code>String[]</code>	<code>getAllFilesFromCurrentDirectory()</code>
<code>String[]</code>	<code>getAllFilesFromDirectory(String directory)</code>
<code>String</code>	<code>getCell(CSVFile table, int idxLine, int idxColumn)</code>
<code>CSVFile</code>	<code>loadCSV(String filename)</code>
<code>CSVFile</code>	<code>loadCSV(String filename, char separator)</code>
<code>void</code>	<code>saveCSV(String[][] content, String filename)</code>
<code>void</code>	<code>saveCSV(String[][] content, String filename, char separator)</code>