Haute École de Bruxelles-Brabant
École Supérieure d'Informatique
Bachelor en Informatique

2018 – 2019
DEV1

**DEV1 – Laboratoires d'environnement**

# TD 03 – L'éditeur de texte nano

## Objectifs

Ce TD a pour objectif de vous familiariser avec l'éditeur `nano` à travers l'édition de vos premiers programmes simples.

## Table des matières





Version du document
12 septembre 2018

# 1 TD3 - L'éditeur de texte nano

**Consignes**

- ▷ Ce TD est accompagné d'exercices à faire **avant** de venir au laboratoire.
- ▷ Prenez bien note des réponses aux exercices ainsi que de la façon dont vous avez trouvé ces réponses.

## 1.1 La ligne de commande

**Dans votre répertoire home, créez un répertoire td3. Ce répertoire contiendra tous les programmes Java que vous écrirez aujourd'hui. Désormais, vous prendrez cette habitude pour chaque td.**

Parfois, vous devez entrer une commande assez longue parce que les noms de fichiers sont longs et/ou nombreux. Linux offre plusieurs facilités pour simplifier l'entrée de longues commandes.

Tutoriel 1 **La complétion de la commande**

Lorsque vous appuyez sur la touche `TAB`, le shell tente de compléter le début de commande que vous avez déjà tapé. Si plusieurs possibilités existent, elles sont affichées si vous appuyez 2x sur `TAB`.

Supposons que vous ne vous rappeliez plus très bien de la commande qui permet de modifier le mot de passe. Vous vous rappelez juste qu'elle commence par `pas`.

- ✎ Tapez `pas` puis appuyez 2x sur la touche `TAB`.
- ✎ Entrez un `s` puis appuyez à nouveau sur `TAB`.

Exercice 1 **La complétion des noms de fichiers**

La touche de tabulation permet également de compléter un nom de fichier.

1. Dans votre dossier td3, copiez le fichier `monfichieraunomtellementlongquilmeparaitpeuprobabledeletaper2xsanserreur` qui se trouve dans le dossier `/eCours/java/td/td3`.
2. Affichez le contenu de ce fichier en évitant de retaper son nom.

Exercice 2 **Joker**

Le point 20 du guide visuel parle des jokers ; c'est le moment de les utiliser :)

1. Copiez dans votre répertoire td3 créé ci-dessus tous les fichiers du répertoire `/eCours/java/td/td3` dont l'extension est `.java` (c'est possible sans passer par un `cd /eCours/java/td/td3`)
2. Copiez dans votre répertoire td3 tous les fichiers du répertoire `/eCours /java/td/td3` dont la deuxième lettre est un '`x`'.
3. Listez le contenu des répertoires des étudiants (pour rappel, les répertoires des étudiants sont ceux qui se trouvent dans `/home` et qui commencent par un '`g`').
4. Listez le contenu des répertoires des professeurs (pour rappel, les répertoires des professeurs sont ceux qui se trouvent dans `/home` et qui sont composés de 3 lettres).

#### Revenir à une commande précédente

Il arrive souvent qu'il faille entrer une commande qu'on a déjà écrite il y a peu (ou en tout cas fort proche de ce qu'on a déjà écrit). C'est là que les flèches viennent à notre secours.

La flèche vers le haut permet de revenir aux commandes précédentes et de les modifier. À utiliser sans modération...

## 1.2 Un programme correct

Tutoriel 2

### Compiler / exécuter un programme

Commençons par un programme correct et tentons de l'exécuter.

Le fichier `/eCours/java/td/td3/Ex.java` contient un petit programme Java tout simple qui affiche un message de bienvenue. Nous allons l'exécuter.

- Normalement, vous avez déjà copié ce fichier chez vous dans un dossier td3.
- Lisez-le et voyez si vous devinez ce qu'il fait.
- Compilez-le. À quoi correspond cette phase ? Quel(s) est (sont) le(s) fichier(s) créé(s) ?
- Exécutez-le.

## 1.3 Configurer l'éditeur

Un éditeur de texte, même simple comme nano, peut apporter quelques facilités dans l'écriture de programmes.

Tutoriel 3

### Coloration sytaxique

La coloration syntaxique signifie utiliser des couleurs pour mettre en évidence certaines parties d'un code : mots clés, constantes...

Pour utiliser cette facilité, il faut configurer nano. Cette configuration se fait dans le fichier `~/.nanorc`

- Tapez la commande : `nano ~/.nanorc`
- Ajoutez-y la ligne : `include "/usr/share/nano/java.nanorc"`
- Quittez l'éditeur.
- Ouvrez le fichier `Ex.java`. Il devrait être coloré.

Tutoriel 4

### Numérotation des lignes

Nano peut également indiquer le numéro de la ligne sur laquelle se trouve le curseur, ce qui sera pratique pour corriger vos erreurs. Pour cela, il existe deux méthodes.

- Lancez nano avec l'option `-c` : `nano -c monFichier`.
- Dans l'éditeur, appuyez sur `CTRL-c`.

### Indentation

Pour qu'un programme soit lisible, il doit être *indenté*. Ce qui serait pratique lorsqu'on code ce serait qu'un retour à la ligne positionne automatiquement le curseur de façon à être aligné avec la ligne précédente. Pour que nano fasse ça pour nous, il suffit d'ajouter ceci à son fichier de configuration : `set autoindent`.

### Autres configurations

Si vous désirez connaitre d'autres possibilités de configuration de nano, vous pouvez lire le manuel : `man nanorc`.

Pour une *quick ref* en ligne, consultez (par exemple) :
http ://www.codexpedia.com/text-editor/nano-text-editor-command-cheatsheet/
(www.codexpedia.com/text-editor/nano-text-editor-command-cheatsheet/)

## Tutoriel 5 Expérience

- Allez dans votre *home* et tapez la commande : `nano .nanorc`.
  Vous pouvez sortir de l'éditeur.
- Tapez la commande : `ls`. Vous ne voyez pas le fichier `.nanorc` alors qu'il est bien présent puisque vous pouvez l'éditer ! ?
  Il s'agit en fait d'un fichier `caché`.

## Exercice 3 Les fichiers cachés

En Linux, tous les fichiers qui commencent par un point sont considérés comme *cachés*. Par défaut, il ne sont pas montrés à l'utilisateur. C'est surtout utilisé pour des fichiers de configuration. Pour tout de même visualiser un fichier caché, il faut utiliser une option de la commande `ls`.

- ▷ Consultez la page de manuel de la commande `ls` et trouvez l'option qui permet de voir les fichiers cachés.

## 1.4 Mon premier programme

Vous allez à présent écrire votre premier programme de bout en bout. Pour ne pas déroger à la tradition, ce sera le «Hello-World ! »(La plupart des manuels de présentation d'un langage commencent par ce grand classique !).

Vous devez maintenant être capable de faire ça tout seul, sans qu'on vous donne la réponse. Mais si vous avez un problème, vous pouvez toujours appeler votre professeur :)

**Tutoriel 6** **Hello world**

- Écrivez votre premier programme nommé `Hello.java` qui affiche le message «Hello, world ! ».
- Compilez-le et exécutez-le.

## 1.5 Détecter et corriger les erreurs

Jusque là, vous avez pu travailler avec un programme correct. Voyons maintenant comment procéder lorsque des erreurs apparaissent (et ça ne manquera pas d'arriver !)

### Comprendre une erreur de compilation

Un rôle important de la compilation est de repérer les erreurs lexicales et/ou syntaxiques que recèle un fichier source et de les décrire le plus clairement possible afin que le programmeur puisse y remédier. Il est important d'apprendre à comprendre ces messages d'erreur.

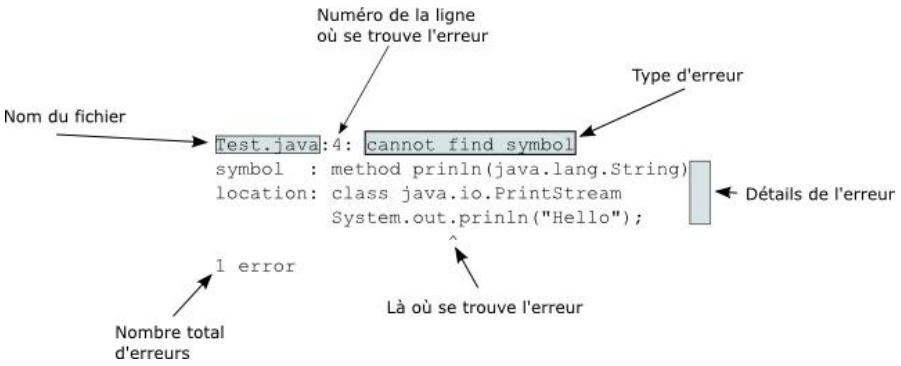


FIGURE 1 – Erreur de compilation

### Comprendre une erreur d'exécution

Lorsqu'on exécute une classe, la machine virtuelle va chercher un fichier ".class" correspondant puis l'exécuter. Il y a donc 2 types d'erreurs bien distincts :

- ▷ soit il ne peut pas trouver le fichier ".class" demandé ;
- ▷ soit il peut trouver une erreur lors de l'exécution de cette classe (division par zéro par exemple).

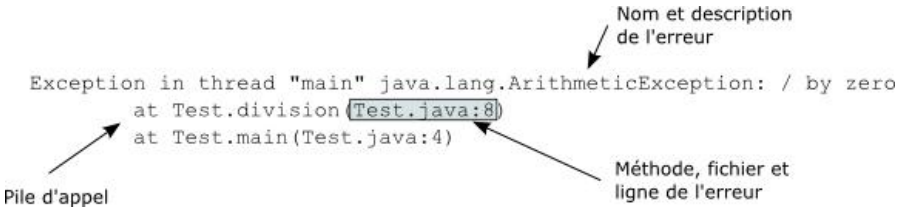


FIGURE 2 – Erreur à l'exécution

Le message d'erreur sera différent ; apprenez à les reconnaitre.

**Note** : la qualité d'un programmeur se voit surtout à sa capacité à réagir face aux erreurs. Prenez note des situations que vous allez rencontrer et des solutions afin de réagir efficacement lorsque vous les rencontrerez à nouveau.

**Exercice 4** **Détection des erreurs**

Vous avez copié dans votre répertoire `td3` tous les fichiers `.java` se trouvant dans `/eCours/java/td/td3`.

Hormis `Ex.java` utilisé précédemment, tous ces fichiers sont erronés. Les erreurs sont tantôt à la compilation, tantôt à l'exécution. Nous vous demandons de détecter et de corriger les erreurs des programmes *Ex1* à *Ex12* et de noter le type d'erreur ainsi que le message engendré.

| Exercice | Erreur | Message affiché |
|---|---|---|
| **Ex1** | | |
| **Ex2** | | |
| **Ex3** | | |
| **Ex4** | | |
| **Ex5** | | |
| **Ex6** | | |
| **Ex7** | | |
| **Ex8** | | |
| **Ex9** | | |
| **Ex10** | | |

## 1.6 Transfert de fichiers

Vous n'avez peut-être pas fini. Pour pouvoir continuer à la maison sans tout recommencer, il serait bon de pouvoir récupérer ce que vous avez déjà fait sur linux1.

Il existe plusieurs méthodes décrites dans l'aide-mémoire (dans le répertoire *aide*).

En voici une.

**Tutoriel 7** **Transférer des fichiers**

- Ouvrez l'explorateur de fichier Windows (par exemple en cliquant sur l'icône "My Computer").
- Dans le champ d'adresse, tapez l'adresse `ftp://linux1`.
- Une boite de dialogue vous demande votre login et mot de passe (sur *linux1*).
- Vous voyez apparaitre un dossier "linux1" qui correspond à votre dossier sur *linux1*.
- Vous pouvez y prendre/déposer des fichiers comme vous le feriez pour un dossier Windows. Vous pouvez par exemple les mettre sur une **clé USB**.