

TP n°1

initiation, structures de contrôle

1 L'environnement de développement

Il existe de nombreux environnements de développement intégré (IDE). Dans cette ressource, nous utiliserons Visual Studio Code sous Linux.

Pour le lancer, il ne faut jamais passer par le menu principal du bureau. Il faut toujours passer par un terminal. C'est ce que nous allons faire maintenant.

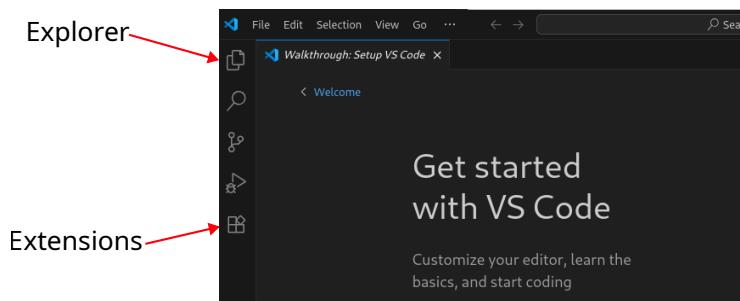
Lancez un terminal en cliquant au bas de l'écran sur l'icône qui représente une fenêtre noire avec un symbole dollar. Pour ne pas tout mélanger, vous allez créer un répertoire spécifique pour la ressource en tapant les commandes qui suivent :

```
mkdir -p ~/BUT1/R1.01
ln -s ~/BUT1/R1.01 ~/BUT1/dev
cd ~/BUT1/dev
```

La première commande crée un répertoire BUT1 dans votre compte (représenté par le symbole ~) et un sous répertoire R1.01 (l'identifiant officiel de la ressource qu'on retrouve sur Moodle). Comme ce nom n'est pas très parlant, la seconde commande crée un lien (autrement dit un alias) pour ce répertoire. De ce fait, les deux noms R1.01 et dev désignent le même sous-répertoire de BUT1. Vous pouvez utiliser l'un ou l'autre à votre convenance. Vous devez prendre l'habitude de placer tous vos fichiers de ce cours dans ce répertoire. La dernière commande permet de se placer dans ce répertoire au niveau du terminal.

Dans ce même terminal, tapez la commande `code &`. Le “et commercial” à la fin de la commande permet de lancer le programme et de continuer à taper des commandes dans le terminal (lancement en tâche de fond).

Au démarrage, commencez par cliquer sur le bouton de gestion des “extensions” (à gauche, voir ci-dessous). Tapez “java” dans la barre de recherche, puis cliquez sur le bouton pour installer “Extension Pack for Java” par Microsoft. Soyez très patient, le téléchargement et l'installation peuvent prendre un certain temps. Il faut attendre tant que le bouton bleu indique “Installing”.



Ensuite, ouvrez le répertoire `~/BUT1/dev` en allant dans le menu File/Open Folder, ou éventuellement en cliquant sur le bouton “Explorer” (à gauche, voir ci-dessus). On vous demandera si vous faites confiance au contenu de ce répertoire, comme vous avez le contrôle de ce contenu, vous pouvez répondre “oui”.

Pour préserver votre droit à la vie privée, il est conseillé de désactiver la télémétrie (File/Preferences/Settings/Application/Telemetry, choisir Off à la place de All).

Créez un nouveau fichier texte en choisissant dans le menu “Fichier” l'action “Nouveau fichier”. Donnez le nom de votre fichier “Prog1.java”. Ce nom de fichier doit commencer par une majuscule et se poursuivre par des lettres ou des chiffres. Il doit se terminer par .java. Saisissez le code ci-dessous dans l'éditeur.

```

import java.io.*;

// ceci est mon premier programme java (ou presque ...)
class Prog1 {
    void run() {
        // c'est ici que vous écrirez votre code en Intro. Dev.
        System.out.println("Hello world!");
    }

    // on indique que le programme doit exécuter le code
    // qu'on a écrit dans la procédure run(). On utilisera
    // toujours ce squelette en Intro. Dev. pour éviter au mieux
    // les objets que ne seront vus qu'au second semestre.
    public static void main(String[] args) {
        new Prog1().run();
    }
}

```

Même sans connaître Java, on voit tout de suite que ce programme ne fait rien d'intéressant. Il a seulement pour but de se familiariser avec le squelette de programme qu'on va utiliser et de voir comment lancer un programme Java.

Vous remarquerez qu'on indique visuellement qu'une ligne du programme est sous le contrôle d'une autre ligne grâce à l'indentation, c'est à dire en décalant vers la droite la ligne en insérant des espaces au début. Vous remarquerez aussi qu'un double slash (//) indique que le reste de la ligne est un commentaire.

Essayez de supprimer le point-virgule à la fin de la procédure run(). Notez que l'éditeur place une marque pour vous signaler une erreur et indique le problème dans une bulle quand vous placez votre souris sur la marque ou dans la liste des problèmes quand on l'affiche ("Affichage/Problèmes"). Rectifiez le code et assurez-vous qu'il n'y a pas d'autre problème à rectifier.

Pour lancer le programme, il faut d'abord s'assurer que l'éditeur ne signale aucun problème (voir ci-dessus). Vous pouvez alors cliquer sur le triangle en haut à droite de votre code pour lancer le programme. Vous pouvez alors voir que cela sauvegarde votre fichier et lance un terminal à l'intérieur de l'éditeur pour exécuter le code.

Il n'y a que les programmeurs qui utilisent un éditeur pour exécuter des programmes. Un fois le programme écrit, on veut pouvoir l'exécuter directement. Le principe est de transformer le code écrit en langage Java en un code qui peut être compris par la machine et exécuté. Cette étape de traduction s'appelle la compilation et le programme qui l'effectue s'appelle un compilateur. Pour compiler et exécuter directement un programme Java simple depuis un terminal, voici ce qu'il faut taper :

```
# compilation et exécution du programme
java Prog1.java
```

Cette commande doit être tapée dans le répertoire qui contient nos fichiers. Pour vous en assurer, vous pouvez taper la commande "ls" pour lister les fichiers présents dans le répertoire courant, ou la commande "pwd" (Print Working Directory) pour voir le nom du répertoire courant. Pour changer de répertoire, utilisez la commande "cd ~/BUT1/dev" vue au début.

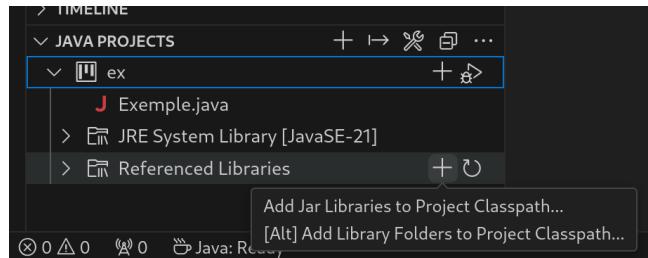
Dans le terminal, quand vous tapez un nom de fichier, vous pouvez vous contenter de taper les premières lettres du nom de ce fichier puis d'appuyer sur la touche tabulation pour compléter automatiquement le nom du fichier (s'il n'y a qu'un choix possible).

Selon les situations, il sera plus simple de lancer vos programmes depuis l'éditeur, ou depuis un terminal. À vous de choisir la meilleure option.

2 Dans l'univers des cartes

Reportez-vous au document sur l'univers des cartes pour savoir comment écrire et exécuter les programmes dans l'univers des cartes.

Dans VSCode, vous avez normalement utilisé 'Open directory' pour ouvrir le répertoire qui doit contenir votre programme Java et le fichier cartes.jar que vous avez récupéré sur Moodle. Dans l'explorateur qui se trouve à gauche de l'écran (voir ci-dessous), dépliez l'item 'Java project', trouvez 'Required libraries', cliquez sur le + à droite de la ligne et ajoutez le fichier cartes.jar.



SÉQUENCE

Une séquence d'instructions permet d'indiquer que des instructions doivent être exécutées les unes à la suite des autres, dans l'ordre indiqué par le programme. Dans le programme, on écrit les instructions dans l'ordre de leur exécution, à raison de une par ligne. Ces instructions doivent toutes être alignées dans le code (même indentation).

```
instruction1;
instruction2;
...
```

Exercice 1 : Écrivez des programmes dans l'univers des cartes qui permettent de passer de la situation initiale donnée à la situation finale indiquée. Les tas de cartes sont ici décrits en donnant en premier la carte la plus au fond du tas et en terminant par la carte au dessus du tas. Indication : dans ces exercices très simples, il n'y a qu'à utiliser l'instruction `deplacer(source, destination)` dans la partie traitement du programme.

1. Situation initiale : 6 de cœur, 7 de cœur, 8 de cœur sur le tas 1
Situation finale : 7 de cœur sur le tas 1, 6 de cœur sur le tas 3, 8 de cœur sur le tas 2
2. Situation initiale : 6 de cœur, 7 de cœur, 8 de cœur sur le tas 1
Situation finale : 8 de cœur, 6 de cœur, 7 de cœur sur le tas 1
3. Situation initiale : 4 cartes sur le tas 1
Situation finale : les 4 cartes dans l'ordre inverse sur le tas 2
4. Situation initiale : 4 cartes sur le tas 1
Situation finale : les 4 cartes dans le même ordre sur le tas 2
5. Situation initiale : 4 cartes sur le tas 1
Situation finale : les cartes réparties sur 4 tas, à raison de une carte par tas
6. Situation initiale : 3 tas, avec sur le tas 1 les cartes suivantes : 5 de pique, 4 de pique, 3 de pique, 2 de pique
Situation finale : les mêmes cartes dans le même ordre sur le tas 2. À aucun moment dans le transfert des cartes on ne doit trouver une carte plus forte posée au dessus d'une carte plus faible. Par exemple, on ne peut pas mettre un 5 au dessus d'un 4, ni d'un 3, ni d'un 2. Le 2 étant la carte la plus faible, il ne doit jamais se trouver sous une autre carte. Décomposez le problème en cherchant des étapes intermédiaires. Par exemple, pour pouvoir déplacer le 5 à sa place finale, il faut que le tas 2 soit vide, et que le tas 3 contienne les cartes 4, 3 et 2 (du fond du tas vers le haut). Trouvez d'autres intermédiaires pour arriver à cette étape. Continuez ainsi jusqu'à ce que les étapes soient évidentes.

FOR, WHILE

En Java, pour répéter n fois une séquence d'instructions, on peut écrire

```
for(int nomDuCompteurDeBoucle = 0; nomDuCompteurDeBoucle < n; nomDuCompteurDeBoucle++) {  
    instruction1;  
    instruction2;  
    ...  
} // fin des instructions répétées
```

Notez que les accolades encadrent les instructions à répéter et que l'indentation permet de mieux visualiser les instructions qui sont répétées.

Si on veut répéter la séquence d'instructions pour toute valeur du compteur de boucle comprise entre debut et fin (bornes comprises), on écrit

```
for(int nomDuCompteur = debut; nomDuCompteur <= fin; nomDuCompteur++) {  
    instruction1;  
    instruction2;  
    ...  
} // fin des instructions répétées
```

En Java, pour répéter une séquence d'instructions tant qu'une condition est vraie, on écrit

```
while (condition) {  
    instruction1;  
    instruction2;  
    ...  
} // fin des instructions répétées
```

Notez que les accolades encadrent les instructions à répéter et que l'indentation permet de mieux visualiser les instructions qui sont répétées.

Exercice 2 : Écrivez des programmes dans l'univers des cartes qui permettent de passer de la situation initiale donnée à la situation finale indiquée. Vous devrez utiliser des boucles.

1. Situation initiale : 16 cartes sur le tas 1

Situation finale : la carte qui était au fond du tas 1 doit être maintenant sur le tas 2, les 15 cartes qui étaient au dessus doivent rester sur le tas 1 dans le même ordre

2. Situation initiale : 16 cartes sur le tas 1

Situation finale : les 16 cartes dans l'ordre inverse sur le tas 2

3. Situation initiale : 16 cartes sur le tas 1

Situation finale : les 16 cartes dans le même ordre sur le tas 2

4. Situation initiale : 6 cartes sur le tas 1

Situation finale : les cartes réparties sur 6 tas, à raison de une carte par tas

5. Situation initiale : un nombre quelconque de cartes sur le tas 1

Situation finale : les mêmes cartes dans l'ordre inverse sur le tas 2

6. Situation initiale : un nombre quelconque de cartes sur le tas 1

Situation finale : les mêmes cartes dans le même ordre sur le tas 2

IF, IF/ELSE

En Java, pour exécuter une séquence d'instructions uniquement quand une condition est vraie, on peut écrire

```
if (condition) {  
    instruction1;  
    instruction2;  
    ...  
} // fin des instructions conditionnelles
```

Notez que les accolades encadrent les instructions sous le contrôle du if et que l'indentation permet de bien visualiser quelles instructions sont contrôlées par le if.

En Java, pour exécuter une séquence d'instructions quand une condition est vraie et une autre séquence quand cette même condition est fausse, on écrit

```
if (condition) {  
    // les instructions 1 et 2 sont exécutées uniquement si la condition est vraie  
    instruction1;  
    instruction2;  
}  
else {  
    // les instructions 3 et 4 sont exécutées uniquement si la condition est fausse  
    instruction3;  
    instruction4;  
} // fin des instructions conditionnelles
```

Notez que l'indentation permet de bien visualiser quelles instructions sont contrôlées par le if et le else.

Exercice 3 : Écrivez des programmes dans l'univers des cartes pour réaliser les opérations ci-dessous. Vous aurez besoins de boucles et de tests.

1. Situation initiale : des cartes rouges sur le tas 1

Situation finale : les coeurs sur le tas 2, les carreaux sur le tas 1

2. Situation initiale : des cartes sur le tas 1

Situation finale : les cartes du dessus du tas 1 transférées sur le tas 2 par groupe de 3, les dernières cartes ne formant pas un groupe de 3 complet restant sur le tas 1.

3. Situation initiale : des cartes sur le tas 1

Situation finale : si la carte au fond du tas 1 est un cœur, toutes les cartes transférées sur le tas 2

4. isoler sur le tas 2 l'ensemble des cartes rouges et l'ensemble des rois du tas 1 et mettre le reste des cartes sur le tas 3

5. répartir le plus équitablement possible le tas de carte 1 en deux tas (2 et 3). Vous n'êtes pas autorisés à compter les cartes.

6. répartir le plus équitablement possible le tas de carte 1 en trois tas (2, 3 et 4). Vous n'êtes pas autorisés à compter les cartes.

7. inverser l'ordre des cartes du tas 1

8. déterminer le plus grand des tas 1 et 2 (celui qui a le plus de cartes). Vous n'êtes pas autorisés à compter les cartes.

9. mélanger les cartes du tas 1 (i.e. répartir les cartes sur deux tas puis les regrouper sur le même tas et répéter ces deux opérations un certain nombre de fois (au moins 10))

Exercice 4 : Écrivez des programmes dans l'univers des cartes pour

1. isoler sur le tas 2 la plus forte carte du tas 1

2. trier les cartes du tas 1 en utilisant la question précédente (tri par sélection)

3. reproduire sur le tas 2 le tas de carte présent sur le tas 1 (les mêmes cartes dans le même ordre) en utilisant une pioche présente sur le tas 3 et dont on suppose qu'elle contient au moins toutes les cartes du tas 1.

4. inverser l'ordre des nations dans un tas composé d'un tas de cartes d'une certaine nation et au dessus d'un tas de cartes d'une autre nation.

5. produire une alternance régulière cœur, carreau, trèfle, pique à partir d'un jeu complet présent sur le tas 1. Combien de tas faut-il pour résoudre le problème ?

6. produire la plus grande alternance régulière possible respectant le motif 4 coeurs, 3 carreaux, 2 trèfles et 1 pique à partir de cartes présentes sur le tas 1

7. Les couleurs : on retourne les quatre premières cartes du jeu sur la table. S'il y a deux cartes de la même famille, on les retire. Aux places vacantes, on pose les deux cartes suivantes du paquet et l'on recommence, jusqu'à ce que l'on ait éliminé toutes les cartes du jeu ou jusqu'à ce que la partie soit bloquée parce qu'il y a quatre familles différentes sur la table.
8. La "Marie-Antoinette" : on répartit le jeu de 32 cartes en 8 tas. On retourne alors la première carte de chaque tas et l'on élimine, deux à deux, les cartes de même valeur. Toute carte éliminée fait apparaître la carte suivante du tas.
9. programmer un tri par insertion
10. programmer un tri à bulle
11. à partir de deux tas déjà triés, créer un tas contenant toutes les cartes et qui soit également trié
12. trouver la plus grande monotonie (suite de cartes consécutives et figurant dans l'ordre croissant) présente dans le tas 1
13. mettre sur un tas l'ensemble des paires, brelans et carrés d'un jeu
14. trouver la plus longue séquence de cartes (suite de cartes consécutives) sans tenir compte de leur couleur
15. même question mais en tenant compte des couleurs
16. compter les points du tas de carte 1 en suivant les règles de la manille. Chaque levée compte pour un point, une manille (le 10) vaut 5 points, l'as vaut 4 points, le roi 3, la dame 2 et le valet 1 point.
17. compter les points du tas de carte 1 en suivant les règles (simplifiées) de la belotte. En atout, un valet vaut 20 points, le 9 vaut 14 points, l'as vaut 11 points, le 10 vaut 10 points, le roi vaut 4 points et la dame vaut 3 points. Dans les autres couleurs, l'as vaut 11 points, le 10 vaut 10 points, le roi vaut 4 points, la dame vaut 3 points et le valet vaut 2 points.

3 Quitter l'ordinateur

Avant de quitter l'ordinateur, il convient de se déconnecter, c'est à dire laisser la place à l'utilisateur suivant. N'oubliez jamais de le faire sinon la prochaine personne passant sur la machine pourrait utiliser votre compte à votre insu mais toujours sous votre responsabilité (elle pourrait d'ailleurs en profiter pour effacer tous vos fichiers).

Si vous vous êtes connectés en mode graphique, il suffit d'aller dans le menu principal (en bas à gauche de l'écran) pour choisir le bouton "Se déconnecter" (le deuxième en partant du bas) et de confirmer. S'il n'y a pas cours dans la salle juste après, vous devez éteindre la machine au lieu de seulement vous déconnecter. Il suffit d'aller dans le menu principal (en bas à gauche de l'écran) pour choisir le bouton "Fermer" (le premier en partant du bas) et de confirmer.

Parfois, on veut quitter temporairement son poste de travail, sans pour autant laisser la place à quelqu'un d'autre. Dans ce cas, il est impératif de verrouiller son écran ("lock screen"). On peut le faire depuis le menu principal, ou avec le raccourci clavier Ctrl-Alt-L. Il devient alors impossible d'utiliser l'ordinateur sans fournir votre mot de passe. Cela interdit donc à une personne mal intentionnée d'accéder à votre compte en votre absence. Si vous oubliez de verrouiller l'écran, on peut facilement pirater votre compte. Vous devez donc impérativement avoir le réflexe de verrouiller votre compte avant même de vous lever de votre chaise !