TP n°1 – Programmes JAVA créés avec un éditeur simple : NotePad++

Remarque : pour pouvoir réaliser ce TP n°1, le Java Development Kit ou JDK (incluant un JRE) doit avoir été installé et paramétré sur le PC et un PATH créé dans l’environnement système (au lycée cela a été réalisé).

# Introduction

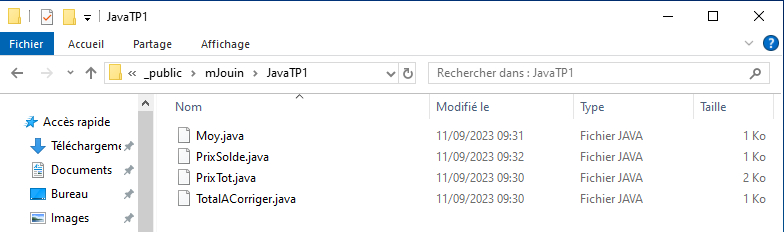
La réalisation avec un éditeur simple a comme particularité que c'est vous qui devez demander la **compilation** et l'**exécution** (les 2 étapes pour exécuter un programme Java) à l'aide de commandes dans une fenêtre spéciale : l'invite de commandes.

Pour les TP suivants, nous utiliserons Eclipse qui est un environnement de développement (EDI) et qui propose une interface graphique qui permet d'exécuter ces 2 commandes en un simple clic sur une icône.

Dans la suite du TP , les actions à exécuter sont précédées par , les autres paragraphes sont là pour donner des explications.

Recopier les programmes du dossier Bloc1Dev\Java\TP1 dans un dossier SIO1\_Public\VotreNom\JavaTP1 (à vous de créer les dossiers).

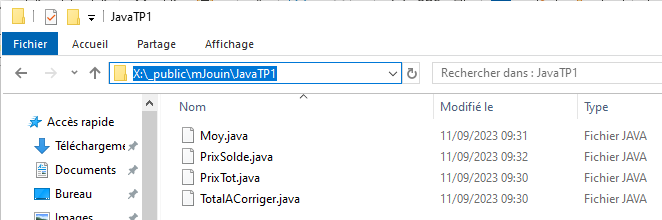
**Laisser la fenêtre de l’explorateur Windows ouverte** sur le contenu de SIO1\_Public\VotreNom\JavaTP1. Voici ce que cela donne pour moi :



# Compiler et exécuter un programme avec l'invite de commandes.

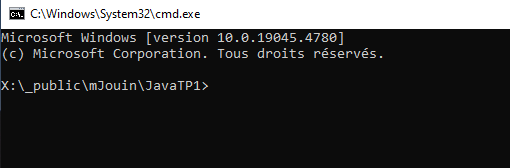
## Ouverture de l'invite de commandes

Dans la fenêtre ouverte précédemment, cliquer à droite dans la ligne qui indique le chemin du dossier cela permet de le sélectionner comme ci-dessous**:**



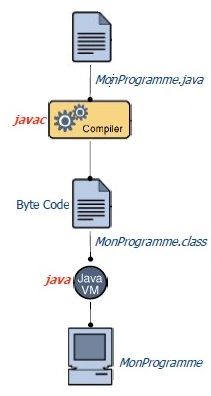
remplacez le chemin sélectionner en saisissant cmd puis validez.

Une nouvelle fenêtre s’ouvre prête à exécuter vos commandes à l’emplacement où sont stockés les programmes ( représenté par X:\\_Public\mJouin\JavaTP1) :



## Fonctionnement d'un programme Java

Un programme Java, pour s'exécuter, a besoin :

* D'être compilé grâce au compilateur javac.exe
* D'être interprété grâce à la machine virtuelle gérée par java.exe
* En Java, les majuscules sont différentes des minuscules (on dit que Java est sensible à la casse). Donc dans les programmes et les noms de fichiers, faites attention !

## Compilation du programme

Lancer la compilation de Moy.java (programme qui fait la moyenne de 3 notes) en entrant la commande *javac Moy.java*

Dans le schéma ci-contre la compilation de MonProgramme.java se fait grâce à l'utilitaire javac.exe.

Si tout se passe bien (pas d'erreur dans le nom du fichier, pas d'erreur dans le code), un fichier Moy.class qui est le résultat de la compilation apparaît (visible grâce à l’explorateur de Windows ou en tapant la commande *dir* dans la fenêtre de l’invite de commande pour voir les fichiers du dossier courant).

## Exécution du programme

Exécution du programme en entrant la commande *java Moy*

Le programme vous demande de saisir une note.

Répondez en utilisant la virgule (pas le point !) si la note est un nombre à virgule. Exemple : saisir 12,5

# Ouvrir un programme

Observez le code source de ce programme en l'ouvrant avec notePad++ : clic droit sur le nom du fichier pour afficher le menu contextuel et choisissez *Ouvrir avec notePad++.*

## Contenu du programme et explications

Un programme Java est toujours stocké dans une classe dont le nom commence par une majuscule (ici Moy). Le nom de la classe doit être le même que le nom du fichier (qui est suivi par .java). Le contenu de la classe est délimité par des accolades { et } ici surlignées en bleu.

Un programme Java contient toujours une méthode principale nommée main() qui contient les instructions du programme (ce qu'il doit faire). Son contenu est délimité par des accolades { et } surlignées ici en rose. Si vous souhaitez ajouter des instructions, il faudra le faire entre ces deux accolades.

Les commentaires en vert fournissent des précisions pour permettre à un développeur qui devra faire évoluer le programme de mieux le comprendre.

1. **import** java**.**util**.**Scanner**;** // nécessaire pour saisir les notes
2. public class Moy**{** // un programme est stocké dans une classe ici Moy
3. public static void main **(**String**[]** args**){** // méthode principale
4. // déclaration des 3 notes et de la variable pour le résultat
5. float n1 **,** n2 **,** n3**,** moy**;**
6. Scanner sc **=** **new** Scanner**(**System**.**in**);** // création de sc de la classe Scanner
7. /\* ligne 8 = Affichage d'un message, ligne 9= saisie d'une note grâce à sc de la classe Scanner qui sera stockée dans la variable n1 \*/
8. System**.**out**.**print **(**"Entrez la premiere note : "**)** **;**
9. n1 **=** sc**.**nextFloat**();** // nextFloat() permet la saisie de réel
10. System**.**out**.**print **(**"Entrez la deuxieme note : "**)** **;**
11. n2 **=**sc**.**nextFloat**();**
12. System**.**out**.**print **(**"Entrez la troisieme note : "**)** **;**
13. n3 **=**sc**.**nextFloat**();**
14. //calcul de la moyenne
15. moy**=(**n1**+**n2**+**n3**)/**3 **;**
16. //affichage du résultat
17. System**.**out**.**println**(**"La moyenne est "**+** moy**);**
18. **}**
19. }

Les données saisies par l'utilisateur ainsi que le résultat du calcul doivent être stockés dans des variables que l'on doit déclarer. C'est le but de la ligne 5 où float indique le type de données stocké ici ce sont des notes donc des réels. Les différents types de variables possibles en Java sont détaillés au II.3 du chapitre 2 (int à la place de float indiquerait que ce sont des entiers).

La ligne 6 permet de créer un objet sc de la classe Scanner qui sera utilisé à la ligne 8 pour la saisie d'une note grâce à la méthode nextFloat() (nextInt() est utilisé pour la saisie d'un entier).

La valeur saisie est stockée dans n1 (l'instruction pour le stockage appelé 'affectation' est matérialisée grâce au signe = ).

La ligne 7 System**.**out**.**print **()** sert à afficher un message pour communiquer avec l'utilisateur ici on veut lui indiquer ce qu'il doit faire cad afficher le message "Entrez la premiere note : ".

La ligne 17 est un peu différente : elle affiche un message (encadré par des guillemets) mais le complète par la valeur stockée dans la variable moy grâce au signe + .

La ligne 15 permet de stocker dans moy (grâce au signe =) le résultat du calcul (n1+n2+n3)/3.

Vous remarquez que j'ai ajouté des explications dans le programme grâce à des commentaires qui sont toujours précédés par // ou encadrés par /\* \*/ s'ils occupent plusieurs lignes.

Si vous avez testé vos programmes avec des réels comme 12,2 12,4 et 12,6 vous obtiendrez un résultat un peu étonnant 12,33339. Cela vient du fait que la partie décimale est gérée avec moins d'octets que pour le type double. Donc soit on arrondira, soit on utilisera le type *double* à la place de float OU LES DEUX !

# Exécuter et modifier un programme

Compilez le programme PrixTot grâce à la commande *javac PrixTot.java* et exécutez-le grâce à la commande java *PrixTot*

Tester ce qui se passe si le prix saisi contient des décimales (rappel : utiliser la virgule pour saisir les décimales. Exemple 12,5) ?

Réflexion que vous devez mener :

D’où l’erreur peut-elle provenir ? regarder la ligne indiquée

Regarder le type d’erreur.

Modifier le programme pour que l’on puisse saisir un prix avec des décimales. Enregistrez et compilez votre programme. Vérifiez en exécutant le programme.

**Une mise à jour c'est : 1. Modifier 2. ENREGISTRER 3. Compiler 4. Exécuter**

# Autres applications

Petite mise à jour : On souhaite maintenant que le programme affiche un message du type "Total a payer : 18.6 euros" (j'ai volontairement omis l'accent sur le a et le signe € qui ne peuvent pas être affichés correctement dans l'invite de commande de DOS).

Ouvrez le programme source PrixTot.java avec NotePad++.

Modifiez l'instruction qui gère l'affichage. Enregistrez et compilez votre programme. Puis exécutez-le.

Les soldes arrivent. Pas toujours facile de calculer un prix soldé. Le programme PrixSolde.java permet à partir du prix initial d'un article et du pourcentage de solde de calculer le prix à payer une fois la remise déduite.

Ouvrez le programme source PrixSolde.java avec NotePad++.

En vous aidant des programmes vus précédemment, compléter les commentaires fournis dans le programme par les instructions qui conviennent parmi les suivantes :

mtRemise= prixInit \* (pourcent/100f);

float prixInit;

float pourcent; // exemple 20 pour 20%

System.out.print ("Entrez le pourcentage de la remise : ");

System.out.println("Montant de la remise correspondant a "+pourcent+" % : "+ mtRemise+" euros .");

System.out.print ("Entrez le prix de l'article en euros : ");

sc.close(); // fermeture de l'objet sc dès que les saisies sont terminées

float mtRemise, prixAPayer;

prixInit =sc.nextFloat();

pourcent =sc.nextFloat();

Enregistrer et tester votre programme qui doit afficher le montant de la remise.

Compléter le programme pour qu'il calcule le prix à payer et qu'il l'affiche.

Proposez un test pour vérifier qu'il fonctionne. Prenez le temps d'écrire le test à faire et le résultat attendu AVANT d'exécuter votre programme. Dans un compte-rendu, recopier le tableau ci-dessous et indiquez le test réalisé (j'ai volontairement mis deux lignes, au cas où le premier test ne donnerait pas le résultat attendu).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N° du test | Valeurs à saisir | | Valeurs attendues pour | | Validation  (oui ou non)  Oui si résultats OK |
| prixInit | pourcent | mtRemise | prixAPayer |  |
| 1 | 10 | 5 | 0,5 | 9,5 | oui |
| 2 | 50 | 25 | 12,5 | 37,5 | oui |

# Interpréter les erreurs de compilation

Compiler le programme TotalACorriger.java et capturer les erreurs de compilation indiquées par javac (vous ferez une capture d'écran de chaque message en utilisant l’outil capture d’écran).

Il manque un point virgule

Coller sur un compte rendu le premier message d’erreur qui indique la première ligne à corriger, ainsi que le type d’erreur. Indiquer un commentaire pour expliquer l’erreur.

Ne supprimer pas de ligne dans le programme (éventuellement vous pouvez les mettre en commentaires) pour que les numéros de ligne concernent toujours les mêmes lignes.

Regarder si d’autres erreurs dans les lignes suivantes peuvent être corrigées avant de recompiler pour ré-exécuter. Pensez à coller une capture du message et à indiquer un commentaire pour expliquer l’erreur (le but est d’avoir un aide-mémoire pour vous aider à résoudre les erreurs).

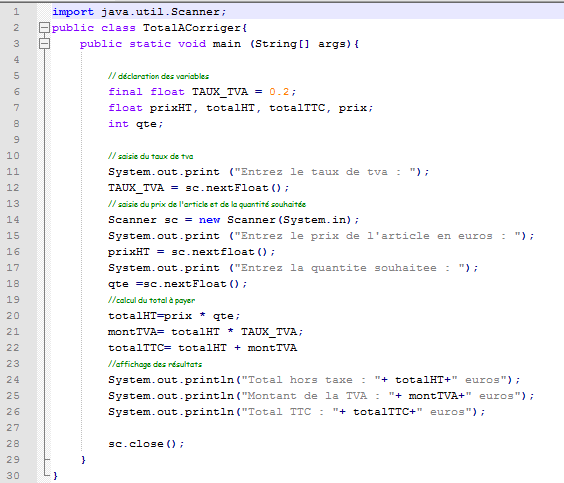
→ On l’a initialisé au mauvais type

→ mettre la ligne : Scanner sc = new Scanner(System.in) avant cette ligne

→ enlever le mot «  final » à TAUX\_TVA

→ la ligne « Scanner sc = new Scanner(System.in) » apparaît 2 fois

Voici le programme :



# Créer et tester un programme (la ou je me suis arreté )

1. Vous préparez un voyage dans l'ouest des Etats Unis avec 3 amis et vous avez constaté que tous les prix sont toujours indiqués en $ et hors-taxe. Par exemple, une chambre avec 2 lits queen-size à l'hôtel 'Premier Inns' de San Diego coûte 50$ (mais il faut rajouter 10,57% de taxes vat et 2% de taxes de séjour).

Les deux taxes ont toutes les deux pour base de calcul le tarif de base de la chambre.

Ces deux taxes sont différentes : chaque état des Etats Unis possède sa propre vat, chaque ville a sa propre taxe de séjour.

L'étape suivante de votre voyage est à Los Angeles et là, une chambre au Budget Inn Hollywood coûte 65$ (et il faut ajouter 10.57% de taxes et 2.50% de taxe de séjour).

Créez un programme qui demande

* le prix d'une chambre en dollars
* le taux de tva (vat aux USA) : exemple 10.57
* le taux de taxe de séjour : exemple 2.50

Le programme calculera le total taxes comprises et l'affichera. Formule de calcul du montant d'une taxe : prix de la chambre \* (taux /100)

Testez le programme et vérifiez que son résultat est cohérent avec chaque exécution suivante :

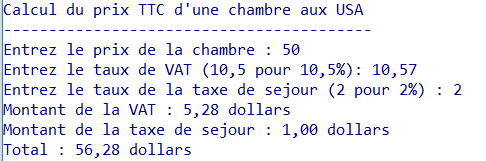


Figure 1 - Total pour San Diego.

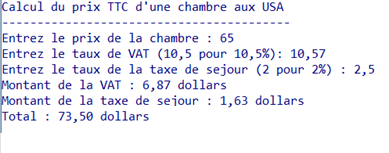


Figure 2 - Total pour Los Angeles.

Amélioration pour les plus avancés :

Complétez le programme pour qu'il vous affiche également le montant correspondant en euros (vérifiez sur Internet le cours du dollar).

Testez le programme et vérifiez que son résultat est cohérent.

En fin de TP, sauvegardez le dossier du TP dans votre espace de stockage personnel.