

## Travail pratique #2

Ce travail doit être fait individuellement.

**Notions mises en pratique** : le respect d'un ensemble de spécifications, les instructions Java, la conception et l'utilisation de méthodes, la classe `String`.

### 1. Spécifications

Il s'agit d'écrire un programme qui permet de construire un horaire de cours, de le modifier, et de le visualiser.

#### 1.1 Lancement du programme

Le programme débute en présentant brièvement le logiciel, puis en offrant un menu de 5 options :

Ce programme permet de construire et modifier un horaire de cours.

```
----  
MENU  
----
```

1. AJOUTER UN COURS A L'HORAIRE
2. SUPPRIMER UN COURS A L'HORAIRE
3. VISUALISER L'HORAIRE
4. EFFACER TOUS LES COURS A L'HORAIRE
5. QUITTER LE PROGRAMME

ENTREZ VOTRE CHOIX :

Vous devez valider le choix au menu. Attention, les seuls choix valides sont 1, 2, 3, 4, ou 5. Par exemple, la valeur 11 ne doit pas être considérée comme valide.

**Note** : au lancement du programme, l'horaire est vide (il ne contient aucun cours ou laboratoire).

#### 1.2 Option 1 : ajouter un cours à l'horaire

Le choix de l'option 1 au menu permet d'ajouter un cours à l'horaire.

1. Le programme demande d'abord d'entrer le sigle du cours à ajouter. Un sigle valide doit être formé d'exactly 7 caractères, dont les 3 premiers sont des lettres (peu importe la casse), et les 4 derniers sont des chiffres. Par exemple, "inf1120" ou "INF1120" ou "Inf1120" est un sigle valide. Vous devez valider le sigle du cours entré.
2. Le programme vérifie ensuite qu'aucun cours ou laboratoire avec ce sigle n'existe déjà dans l'horaire (peu importe la casse).
3. Si un cours/labo ayant ce sigle (peu importe la casse) est présent dans l'horaire, le programme affiche un message à cet effet, en marquant une pause pour laisser le temps à l'utilisateur de lire le message. Comme ceci :

```
*****  
AJOUTER UN COURS A L'HORAIRE  
*****
```

Sigle du cours : **inf1120**

OPERATION ANNULEE.

Ce cours ne peut être ajouté car il est déjà présent dans votre horaire.

Appuyez sur "ENTREE" pour continuer...

Lorsque l'utilisateur saisit la touche « ENTRÉE », l'option se termine, sans ajouter le cours à l'horaire, et le menu principal est affiché de nouveau, dans l'attente d'un autre choix au menu.

4. Si le sigle du cours entré n'existe pas à l'horaire : le programme demande d'entrer 1) le jour, 2) le moment du jour, et 3) le groupe du cours à ajouter (dans cet ordre). Un jour valide est soit "lundi", "mardi", "mercredi", "jeudi", ou "vendredi" (peu importe la casse). Vous devez valider le jour saisi. Un moment du jour valide est soit "am", "pm" ou "soir" (peu importe la casse). Vous devez valider le moment du jour saisi. Un groupe valide est un nombre entier contenant exactement 2 chiffres. De plus, un groupe valide doit respecter les règles suivantes par rapport au jour du cours :

Jour du cours	Groupe valide
Lundi	Le premier chiffre du groupe doit être 1 (exemple 10, 14, ...)
Mardi	Le premier chiffre du groupe doit être 2 (exemple 20, 22, ...)
Mercredi	Le premier chiffre du groupe doit être 3 (exemple 30, 31, ...)
Jeudi	Le premier chiffre du groupe doit être 4 (exemple 40, 43, ...)
Vendredi	Le premier chiffre du groupe doit être 5 (exemple 50, 56, ...)

Vous devez valider le groupe du cours saisi par l'utilisateur.

5. Le programme vérifie ensuite si la plage horaire donnée par l'utilisateur (jour et moment du jour) est disponible (il n'y a aucun cours ou laboratoire inscrit à cette plage horaire).
6. Si la plage horaire n'est pas disponible, le programme affiche un message à cet effet, en marquant une pause pour laisser le temps à l'utilisateur de lire le message. Comme ceci :

```
*****
AJOUTER UN COURS A L'HORAIRE
*****
```

```

Sigle du cours : inf2323
Jour du cours : lundi
Moment du jour de ce cours : soir
Groupe du cours : 12
```

```

OPERATION ANNULEE.
Il existe déjà un cours (ou laboratoire) dans la plage horaire : LUNDI SOIR.
```

```
Appuyez sur "ENTREE" pour continuer...
```

Lorsque l'utilisateur saisit la touche « ENTRÉE », l'option se termine, sans ajouter le cours à l'horaire, et le menu principal est affiché de nouveau, dans l'attente d'un autre choix au menu.

7. Si la plage horaire est disponible, le programme demande à l'utilisateur si ce cours comprend une période de laboratoire. Les deux seules réponses valides sont "oui" ou "non" (peu importe la casse). Vous devez valider la réponse de l'utilisateur.
8. S'il n'y a pas de laboratoire, **le cours est ajouté à l'horaire**, l'option se termine, et le menu principal est affiché de nouveau, dans l'attente d'un autre choix au menu.
9. S'il y a un laboratoire associé à ce cours, le programme demande à l'utilisateur 1) le jour du laboratoire, et 2) le moment du jour de ce laboratoire (dans cet ordre). Vous devez valider le jour, et le moment du jour du laboratoire comme spécifié au point 4.

10. S'il y a un conflit d'horaire (la plage horaire saisie pour le laboratoire est déjà occupée par un cours ou un laboratoire), le programme affiche un message à cet effet, en marquant une pause pour laisser le temps à l'utilisateur de lire le message. Comme ceci :

```
*****
AJOUTER UN COURS A L'HORAIRE
*****

Sigle du cours : inf1111
Jour du cours : mercredi
Moment du jour de ce cours : soir
Groupe du cours : 34
Ce cours comprend-il une periode de laboratoire (oui ou non) ? oui
Jour du laboratoire : lundi
Moment du jour de ce laboratoire : soir

OPERATION ANNULEE.
Il existe deja un cours (ou laboratoire) dans la plage horaire : LUNDI SOIR.

Appuyez sur "ENTREE" pour continuer...
```

Lorsque l'utilisateur saisit la touche « ENTRÉE », l'option se termine, sans ajouter le cours à l'horaire, et le menu principal est affiché de nouveau, dans l'attente d'un autre choix au menu.

11. S'il n'y a pas de conflit d'horaire pour la période de laboratoire, **le cours et son laboratoire associé sont ajoutés à l'horaire**, l'option se termine, et le menu principal est affiché de nouveau, dans l'attente d'un autre choix au menu.

### 1.2.1 Précisions sur le stockage en mémoire de l'horaire

L'horaire doit être stocké dans une variable de type `String`. Chaque cours ou laboratoire doit se trouver sur une ligne différente (dans la chaîne), et doit respecter le format suivant : `lettre|sigle|groupe|jour|moment du jour`.

lettre	Lettre indiquant si c'est un cours ou un laboratoire (C = cours, L = laboratoire)
sigle	Le sigle du cours / laboratoire
groupe	Le groupe du cours / laboratoire
jour	Le jour du cours / laboratoire (lundi, mardi, ..., vendredi)
moment du jour	Le moment du jour du cours / laboratoire (am = matin, pm = après-midi, soir = soirée)

**Note :** Au lancement du programme, l'horaire est vide, et doit donc être représenté en mémoire par une chaîne de caractères vide.

Par exemple, la variable `horaire` suivante représente un horaire bien formé qui contient des cours et les laboratoires associés à certains cours.

```
String horaire = "C|inf1120|23|mercredi|pm\n"
                + "L|inf1120|23|lundi|am\n"
                + "C|his1230|50|vendredi|soir\n"
                + "C|etd2098|10|lundi|soir\n"
                + "L|etd2098|10|jeudi|am\n"
                + "C|cou2323|32|mardi|am";
```

La variable `horaire` ci-dessus représente un horaire contenant les cours et laboratoires suivants :

- INF1120-23 (cours) : mercredi après-midi
- INF1120-23 (labo) : lundi matin

- HIS1230-50 (cours) : vendredi soir
- ETD2098-10 (cours) : lundi soir
- ETD2098-10 (labo) : jeudi matin
- COU2323-32 (cours) : mardi matin

**Note :** L'ordre dans lequel vous inscrivez les cours et laboratoires dans la chaîne de caractères représentant l'horaire est à votre discrétion. La casse choisie est aussi à votre discrétion.

### 1.3 Option 2 : supprimer un cours à l'horaire

Cette option permet de supprimer un cours qui se trouve déjà dans l'horaire.

1. Si l'horaire ne contient aucun cours lorsque l'utilisateur choisit cette option, le programme affiche un message à cet effet, en marquant une pause pour laisser le temps à l'utilisateur de lire le message. Comme ceci :

```
*****  
SUPPRIMER UN COURS A L'HORAIRE  
*****
```

L'horaire ne contient aucun cours.

Appuyez sur "ENTREE" pour continuer...

Lorsque l'utilisateur saisit la touche « ENTRÉE », l'option se termine, sans supprimer de cours à l'horaire, et le menu principal est affiché de nouveau, dans l'attente d'un autre choix au menu.

2. Si l'horaire n'est pas vide, le programme demande d'entrer le sigle du cours à supprimer de l'horaire. Vous devez valider le sigle entré tel que spécifié à la section 1.2, point 1.
3. Si le sigle du cours saisi par l'utilisateur (peu importe la casse) n'existe pas dans l'horaire, le programme affiche un message à cet effet, en marquant une pause pour laisser le temps à l'utilisateur de lire le message. Comme ceci :

```
*****  
SUPPRIMER UN COURS A L'HORAIRE  
*****
```

Sigle du cours : **INF1111**

OPERATION ANNULEE.

Ce cours n'est pas a l'horaire.

Appuyez sur "ENTREE" pour continuer...

Lorsque l'utilisateur saisit la touche « ENTRÉE », l'option se termine, sans supprimer de cours à l'horaire, et le menu principal est affiché de nouveau, dans l'attente d'un autre choix au menu.

4. Si le sigle du cours saisi par l'utilisateur est un cours à l'horaire (peu importe la casse), **le programme retire ce cours de l'horaire** ainsi que son laboratoire associé (s'il y a lieu), et affiche de nouveau le menu dans l'attente d'un autre choix.

Par exemple supposons qu'avant la suppression, l'horaire en cours est représenté en mémoire par la variable `horaire` suivante.

```
String horaire = "C|inf1120|23|mercredi|pm\n"  
                + "L|inf1120|23|lundi|am\n"  
                + "C|his1230|50|vendredi|soir\n"  
                + "C|etd2098|10|lundi|soir\n"  
                + "L|etd2098|10|jeudi|am\n"  
                + "C|cou2323|32|mardi|am";
```

Si l'utilisateur demande la suppression du cours INF1120, la variable horaire devrait ressembler à ceci (après la suppression) :

```
String horaire = "C|his1230|50|vendredi|soir\n"
                + "C|etd2098|10|lundi|soir\n"
                + "L|etd2098|10|jeudi|am"
```

#### 1.4 Option 3 : visualiser l'horaire

Cette option permet de visualiser l'horaire sous la forme d'un tableau dans lequel les jours sont sur l'axe vertical et les moments du jour, sur l'axe horizontal. Lorsque cette option est choisie, le programme affiche l'horaire, en marquant une pause pour laisser le temps à l'utilisateur de regarder son horaire. Lorsque l'utilisateur saisit la touche « ENTRÉE », l'option se termine, et le menu principal est affiché de nouveau, dans l'attente d'un autre choix au menu. Par exemple, si l'horaire est vide (il ne contient aucun cours) le programme affiche ceci :

```
*****
HORAIRE
*****
```

	MATIN	APRES-MIDI	SOIREE
LUNDI			
MARDI			
MERCREDI			
JEUDI			
VENDREDI			

Appuyez sur "ENTREE" pour continuer...

Si l'horaire contient les cours suivants :

```
String horaire = "C|inf1120|23|mercredi|pm\n"
                + "L|inf1120|23|lundi|am\n"
                + "C|his1230|50|vendredi|soir\n"
                + "C|etd2098|10|lundi|soir\n"
                + "L|etd2098|10|jeudi|am\n"
                + "C|cou2323|32|mardi|am";
```

Le programme affichera ceci :

```
*****
HORAIRE
*****
```

	MATIN	APRES-MIDI	SOIREE
LUNDI	INF1120-23 (LABO)		ETD2098-10 (COURS)
MARDI	COU2323-32 (COURS)		
MERCREDI		INF1120-23 (COURS)	
JEUDI	ETD2098-10 (LABO)		
VENDREDI			HIS1230-50 (COURS)

Appuyez sur "ENTREE" pour continuer...

### 1.5 Option 4 : effacer tous les cours à l'horaire

Cette option permet d'effacer tous les cours qui se trouvent dans l'horaire.

1. Si l'horaire ne contient aucun cours lorsque l'utilisateur choisit cette option, le programme affiche un message à cet effet, en marquant une pause pour laisser le temps à l'utilisateur de lire le message. Comme ceci :

```
*****  
EFFACER TOUS LES COURS A L'HORAIRE  
*****
```

L'horaire ne contient aucun cours.

Appuyez sur "ENTREE" pour continuer...

Lorsque l'utilisateur saisit la touche « ENTRÉE », l'option se termine, et le menu principal est affiché de nouveau, dans l'attente d'un autre choix au menu.

2. Si l'horaire n'est pas vide, avant d'effacer tous les cours, le programme demande une confirmation, comme ceci :

```
*****  
EFFACER TOUS LES COURS A L'HORAIRE  
*****
```

Voulez-vous vraiment supprimer tous les cours de cet horaire (OUI ou NON) ?

Les seules deux réponses valides sont "oui" et "non" (sans tenir compte de la casse). Vous devez valider la réponse de l'utilisateur comme expliqué à la section 1.2, point 7.

3. Si la réponse de l'utilisateur est oui, le programme **supprime tous les cours** de l'horaire, et le menu principal est affiché de nouveau, dans l'attente d'un autre choix au menu.
4. Si la réponse de l'utilisateur est non, le programme ne supprime aucun cours, et affiche un message comme quoi l'opération est annulée, en marquant une pause pour laisser le temps à l'utilisateur de lire le message. Comme ceci :

```
*****  
EFFACER TOUS LES COURS A L'HORAIRE  
*****
```

Voulez-vous vraiment supprimer tous les cours de cet horaire (OUI ou NON) ? **non**

OPERATION ANNULEE.

Appuyez sur "ENTREE" pour continuer...

Lorsque l'utilisateur saisit la touche « ENTRÉE », l'option se termine, et le menu principal est affiché de nouveau, dans l'attente d'un autre choix au menu.

### 1.2 Option 5 : quitter le programme

Lorsque l'utilisateur choisit l'option 5, le programme se termine en affichant le message de fin suivant :

FIN NORMALE DU PROGRAMME.

## 2. Précisions supplémentaires

- Vous devez écrire votre programme dans une classe nommée `TP2` qui contiendra votre méthode `main` et toutes les autres méthodes que vous aurez conçues.

- Toutes les méthodes doivent être `public static`.
- Prenez soin de bien modulariser votre code à l'aide de méthodes cohésives et robustes. Voici quelques lignes directrices pour vous guider dans la conception de vos méthodes :
  - Chaque fois que vous vous apercevez que différentes parties de votre code se ressemblent énormément, essayer d'en faire une méthode bien paramétrée (si possible).
  - Faites une méthode pour chaque petite tâche spécifique à accomplir dans la résolution du problème.
  - Généralement (pas tout le temps), une méthode ne devrait pas faire plus de 20-25 lignes de code (y compris la méthode main). Si une méthode semble trop longue, posez-vous la question à savoir si elle pourrait être subdivisée en plusieurs autres méthodes plus spécialisées (cohésives).
- Lorsqu'on vous demande de valider une valeur saisie par l'utilisateur, cela sous-entend une BOUCLE de validation qui sollicite la valeur, lit la valeur, et lorsque la valeur saisie est invalide, elle affiche un message d'erreur, elle sollicite et lit de nouveau la valeur, et ce tant que la valeur saisie n'est pas valide.
- Vous DEVEZ utiliser la classe `Clavier.java` pour effectuer toutes les saisies.
- Les seules CLASSES PERMISES sont `String`, `Math`, `Character`, `Clavier`, et `System`.
- Vous NE DEVEZ PAS utiliser les tableaux.
- Vous NE DEVEZ PAS utiliser les expressions régulières.
- Vous DEVEZ respecter à la lettre les **informations**, les **messages**, et le **format** de l'affichage qui sont montrés dans les exemples d'exécution fournis avec l'énoncé du TP.
- Votre programme NE DOIT JAMAIS PLANTER lors d'une saisie faite par l'utilisateur (utiliser des `String`).
- Toute **VARIABLE GLOBALE** est **INTERDITE** : **toutes les variables doivent être déclarées à l'intérieur d'une méthode** (sauf pour les boucles `for`).
  - Pour passer une information du programme appelant (celui qui appelle la méthode) au programme appelé (la méthode appelée), vous devez passer cette information en paramètres.
  - Pour passer une information du programme appelé (une méthode quelconque) au programme appelant (celui qui appelle la méthode), utilisez la valeur de retour de la méthode appelée.
- Utilisez des constantes (`final`) autant que possible.
- Les constantes doivent être déclarées et initialisées au niveau de la classe : `public static final...`
- L'affichage des résultats doit se faire à la console.
- Utilisez des variables réelles uniquement lorsque requis. Par exemple, un compteur ne doit pas être `float` ou `double`.
- Votre code doit compiler et s'exécuter avec le JDK 7.
- Vous DEVEZ respecter le style Java vu en classe.
- **Le non-respect de toute spécification ou consigne se trouvant dans l'énoncé du TP (et les exemples d'exécution donnés avec l'énoncé du TP) est susceptible d'engendrer une perte de points.**

**Note :** *Si quelque chose est ambigu, obscure, s'il manque de l'information, si vous ne comprenez pas les spécifications, si vous avez des doutes... vous avez la responsabilité de vous informer auprès de votre enseignante.*

### 3. Détails sur la correction

---

#### 3.1 La qualité du code (40 points)

Concernant les critères de correction du code, lisez attentivement le document “critères généraux de correction du code Java” dans la section CONTENU DU COURS / Travaux, sur Moodle.

**Note :** *Votre code sera corrigé sur la totalité ou une partie des critères de correction indiqués ci-dessus, ainsi que sur le respect des spécifications/consignes mentionnées dans ce document.*

#### 3.2 L'exécution (60 points)

**Un travail qui ne compile pas se verra attribuer la note 0 pour l'exécution.**

**Note :** *Votre code sera testé en tout ou en partie. Les points seront calculés sur les tests effectués. Ceci signifie que si votre code fonctionne dans un cas x et que ce cas x n'est pas testé, vous n'obtiendrez pas de points pour ce cas. Il est donc dans votre intérêt de vous assurer du bon fonctionnement de votre programme dans tous les cas possible.*

### 4. Date et modalité de remise

---

#### 4.1 Remise

**Date de remise :** 18 novembre 2016 avant minuit (ou le 19 novembre À MINUIT).

**Le fichier à remettre :** TP2.java (PAS dans une archive zip, rar, etc.)

**Remise via Moodle uniquement.**

Vous devez remettre (téléverser) votre fichier sur le site du cours (Moodle). Vous trouverez la boîte de remise dans la section **BOÎTES DE REMISE - TRAVAUX PRATIQUES / REMISE DU TP2**

#### 4.2 Politique concernant les retards

Une pénalité de 10% de la note finale, par jour de retard, sera appliquée aux travaux remis après la date limite. La formule suivante sera utilisée pour calculer la pénalité pour les retards :  $\text{Nbr points de pénalité} = m / 144$ , où  $m$  est le nombre de minutes de retard par rapport à l'heure de remise. Ceci donne 10 points de pénalité pour 24 heures de retard, 1.25 point de pénalité pour 3 heures, etc.

Aucun travail ne sera accepté après **1 jour** de retard et la note attribuée sera 0.

#### 4.3 Remarques générales

- Aucun programme reçu par courriel ne sera accepté.
- Vous avez la responsabilité de conserver des copies de sauvegarde de votre travail (sur disque externe, Dropbox, Google Drive, etc.). La perte d'un travail due à un vol, un accident, un bris... n'est pas une raison valable pour obtenir une extension pour la remise de votre travail.
- **N'oubliez pas d'écrire votre nom complet et votre code permanent dans l'entête de la classe à remettre.**
- N'attendez pas à la dernière minute pour commencer le travail, vous allez fort probablement rencontrer des problèmes inattendus!

Le règlement sur le plagiat sera appliqué sans exception. Vous devez ainsi vous assurer de ne pas échanger du code avec des collègues ni de laisser sans surveillance votre travail au laboratoire. Vous devez également récupérer rapidement toutes vos impressions de programme au laboratoire.

BON TRAVAIL !