

Consignes. Avant toute chose, prenez conscience que les corrections sont effectuées à l'aide d'un automate, par conséquent toute violation des règles conduira inévitablement à la délivrance d'une note nulle.

- Comment livrer mon code ?
Si mon nom de famille est **yunès** et que je souhaite rendre le travail pour le tp 3 alors je dois construire une archive au format ZIP de nom **yunes-tp3.zip**, le nom ayant été converti en minuscules, sans accent et sans espace (*ie* : **de Gaulle** doit être traduit en **degaulle**). Si vous avez un homonyme rajoutez des initiales, par exemple **yunesjb-tp3.zip**. Respectez votre nommage pour tout le semestre. Vous devez déposer votre code dans le cours Moodle adéquat (je vais en demander l'ouverture, sinon on verra comment faire).
- Que mettre dans l'archive ?
Le code source (et uniquement le code source) correspondant à l'exercice.

De plus, suivez bien les instructions complémentaires des livrables.

Exercice 1

1. Écrire un programme permettant de déterminer le jour de la semaine correspondant au premier jour de l'année 2000 (il s'agit d'un samedi).
2. Modifier le programme de sorte à obtenir le résultat en chinois et en allemand.
3. Modifier le programme de sorte que l'utilisateur puisse choisir l'année et la langue en ligne de commande (la langue devra être spécifiée au format standard des locales, ex.: `fr`) : `java Jour [-y année] [-l langue]`. Si l'année n'est pas spécifiée, il s'agira de l'année courante et la langue celle par défaut. L'affichage du jour de la semaine doit être celui «par défaut» c'est-à-dire par exemple : `FRIDAY`.

Livable : l'archive contenant le programme `Jour.java` (pas de package!).

Exercice 2

1. Écrire un programme permettant d'obtenir la liste de tous les vendredi 13 d'une année donnée en ligne de commande : `java Friday13 [-y année]`. S'il n'y a pas d'argument, il doit s'agir de l'année courante. Le format de sortie doit être : au format : `mois` où le mois s'affiche en anglais et intégralement (pas de raccourci), pas d'espace ni avant, ni après, pas de ligne vide ni avant, ni après. Exemple pour 2019 :

```
September
December
```

Livable : l'archive contenant le programme `Friday13.java` (pas de package!).

Exercice 3

Préliminaire : on définit le jour de paye d'un mois donné comme le 25; sauf lorsque ce jour tombe un samedi ou un dimanche, auquel cas il doit s'agir du vendredi qui précède.

1. Écrire un programme permettant d'obtenir la liste de tous les jours de paye d'une année donnée en argument : `java Paye [-y année]` où l'année par défaut est l'année courante. La sortie doit être au format suivant :

```
25 January 2000
25 February 2000
24 March 2000
25 April 2000
25 May 2000
23 June 2000
```

```
25 July 2000
25 August 2000
25 September 2000
25 October 2000
24 November 2000
25 December 2000
```

Livrable : l'archive contenant le programme `Paye` (pas de package!).

Exercice 4

1. Écrire un programme permettant de calculer le nombre de jour compris entre deux dates indiquées au format `uuuu-MMMM-dd` dans une langue choisie et spécifiée par une locale : `java Nombre -d1 date1 -d2 date2 [-l locale]` où le nombre de jour est calculé entre `date1` et `date2`, et la locale par défaut est `fr`. On pourra supposer que la `date2` est postérieure à `date1`.

Livrable : l'archive contenant le programme `Nombre.java` (pas de package!).