

Ce dernier TP permet de faire un bilan de l'U.E. *Fiabilité 1* et reprend l'ensemble des notions vues en cours.

1 Travail à effectuer

Ce Tp est à rendre par groupe de 2 à 4 étudiants avant le 3 novembre 10h. Le rendu comprend vos **sources** java (tests compris), et un **rapport** en pdf répondant aux différentes questions (et pouvant contenir des approfondissements, et/ou explications techniques). Ce travail sera noté et comptera pour 60% pour votre note de TP de l'U.E. Comme il contient un travail « non nulle » de développement, il est intéressant de se partager/distribuer les tâches de travail dans le groupe. Enfin, ce travail est un bon exercice pour réviser pour l'examen du jeudi 3 novembre 10h.

2 Classe Date

Le but de ce TP est de donner une traduction de la classe python date du module datetime en Java. Le but est bien entendu de **ne pas** utiliser l'api **Date** de java.

L'ensemble des fonctions de l'objet *date* est à implémenter, à l'exception de l'attribut de classe `date.resolution`, les méthodes `date.timetuple()`, `date.strftime(format)` et `date.__format__(format)`, ainsi que la surcharge des opérateurs (non disponible en java).

L'ensemble des fonctionnalités attendues est donc le suivant :

- Accès aux attributs de classe **min**, **max**.
- Accès aux attributs d'instance **year**, **month**, **day**.
- Méthodes **replace(year, month, day)**, **toordinal()**, **weekday()**, **isoweekday()**, **isocalendar()**, **isoformat()**, **__str__()**, **ctime()**.

La spécification de chaque méthode est donnée par la page de documentation python3 <https://docs.python.org/3.3/library/datetime.html#datetime.date>. Pour information, le calendrier grégorien proleptique est le calendrier grégorien (celui actuellement en vigueur dans la grande majorité du monde) étendu de sorte que toute date passée et future appartient aussi au calendrier grégorien. Dans ce calendrier, seules les années multiples de 4 mais non multiple de 100, et les années multiples de 400 sont bissextiles (contiennent un 29 février). Enfin, une méthode simple pour calculer le jour de la semaine est donnée dans l'article wikipédia *Determination of the day of the week*.

Votre travail si vous l'acceptez¹ est le suivant :

1. Écrivez l'interface publique de votre classe java **Date** en transcrivant les fonctionnalités de l'objet python **date** en java. Vous prendrez soin à suivre les conventions usuelles de nommage pratiquées en java, et à convertir les *Error* python en *Exception* java équivalente, quitte à créer des exception spécifiques si nécessaires.
2. Selon le principe du test en **boîte noir**, pour chaque fonction publique de votre classe, déterminez les classes d'équivalences permettant de définir une suite de test. Donnez une suite de tests et valeurs attendues en retour correspondant aux classes d'équivalences que vous avez choisies (pensez aux bornes lors du choix des valeurs). Vous pouvez utiliser la commande unix `date -u -d @timestamp` pour vérifier les résultats attendus de vos tests.
3. Implémentez vos tests sous **JUnit** et votre classe java **Date**. Selon le principe du *Test Driven Development*, il est intéressant d'avoir un retour direct des tests en ajoutant les fonctionnalités attendues au fur et à mesure de l'implémentation des tests. Votre classe java doit être la seule classe testée, vous *mockera* avec *jmockit* les comportements *externes* ou *non-déterministes*.
4. Une fois que tous vos tests passent, notez et analysez le taux de couverture donné par l'outil *Emma*. Selon le principe du test en **boîte blanche**, ajoutez des tests si nécessaires afin d'augmenter votre taux de couverture. Dans votre rapport, vous ferez apparaître ces cas de tests comme *tests ajoutés afin d'augmenter la couverture*.
5. Ajoutez des annotations java de l'outil *Checker Framework* (fiche analyse statique de code java) pour augmenter davantage la fiabilité de votre classe. Modifiez votre classe si les résultats du *checker* ne vous conviennent pas. Terminez par une **analyse statique** de votre programme par un outil suggéré dans la fiche prévue à cet effet, et commentez le résultat.

3 Rendu

Après avoir vérifié que le nom des membres du groupes apparait dans votre travail, vous déposerez votre fichier **.zip** dans le dossier correspondant au cours sur Amétice (ouverture prochaine).

Dans le cas où le dépôt amétice n'est pas possible, vous enverrez vos fichiers par mail à `romaric.duvignau@lif.univ-mrs.fr`.

¹sachant que vous n'avez pas le choix si vous voulez être noté.