

Date de l'examen intra : **Jeudi 17 octobre de 16h30 à 18h45 au
Z-310 Pavillon Claire-McNicoll**

Toute documentation papier (non électronique) est autorisée : par exemple, les livres, notes de cours, codes des travaux pratiques imprimés sont permis.

Le contenu et la matière du TP #2 font également partie de l'examen intra bien que la date de remise soit après l'examen intra.

Absence motivée aux examens : <https://desi.umontreal.ca/reglements/>

Profitez des périodes de démonstration pour avancer dans la réalisation de vos travaux pratiques à l'aide de vos auxiliaires d'enseignement.



Modalités :

À remettre, au plus tard, **mercredi 13 novembre 2024 à 23h59**

Le travail est corrigé sur **35 points**.

Les trois premiers jours de retard entraînent une pénalité de **5 points (sur 35) par jour**. Le jour suivant fait perdre **10 points (sur 35)**.

Les travaux se font seul ou en équipe de deux, au maximum.

Vous ne remettez alors qu'un seul travail.

!/ ATTENTION !/ L'utilisation d'outils de génération de code par intelligence artificielle, et la copie de code d'une autre source, autre que ce cours IFT 1170, seront sévèrement **sanctionnées**.



Remise des travaux : même procédure que le TP #1 (voir l'énoncé au besoin).

NOTE : Il est très important de compléter le TP #2 le plus rapidement possible, quitte à l'améliorer plus tard pour la remise du 13 novembre. Les matières du TP #2 représentent une partie importante de l'examen intra.

Numéro A (15 points)

Matières étudiées : polymorphisme, **this**, **toString**, fichier de type texte, etc.

On dispose de la classe incomplète **Employe** suivante :

```
class Employe
{
private String NAS ; // numéro d'assurance sociale, exemple "123 456 789"
private double salHebdo ; // salaire hebdomadaire
. . . Constructeurs à écrire . . .
. . . Autres méthodes à écrire . . .
} // fin de la classe Employe
```

Soit **TestEmploye**, la classe contenant la méthode **main**.

1) Écrire les constructeurs utilisant le pointeur **this** pour instancier les 4 employés suivants :

```
Employe emp1 = new Employe("250 642 753", 1234.56),
emp2 = new Employe("123 456 789"), // par défaut 1250.25$
emp3 = new Employe("250 343 678", 40.00, 25.75),
// salHebdo = 40 heures par semaine x taux d'horaire 25.75$
emp4 = new Employe("450 279 321", 1750.75);
```

2) Écrire la méthode **afficher** appropriée, puis nous l'utiliserons pour afficher les informations de **emp3** et **emp4** dans la console.

3) Modifier le salaire de **emp2** : il va devenir la somme de salaires de **emp1** et **emp3**. Nous affichons les informations de **emp2** dans la console.

4) Écrire la déclaration et l'initiation d'un tableau de 6 employés dont le contenu est le suivant :

i	emp[i]	
0	"250 642 753"	1234.25\$
1	"123 456 789"	1250.25\$ (par défaut)
2	"250 343 654"	40 hrs au taux de 17.25\$ de 1'heure
3	"231 467 890"	1671.50\$
4	"478 343 689"	1750.75\$
5	"371 238 432"	50 hrs au taux de 20.25\$ de 1'heure

Afficher le contenu du tableau à votre choix, en utilisant la redéfinition de **toString()** ;

5) Écrire la méthode et leurs appels, pour compter puis afficher le nombre d'employés :

a) Qui gagnent moins de 1300.00\$ par semaine dont le **NAS** contient '5'

b) Qui gagnent moins de 750.00\$ par semaine dont le **NAS** contient '3'

6) Trier le tableau (avec la méthode du tri par sélection) selon les **NAS** de ces 6 employés.

7) Créer le fichier texte "**empTri.txt**" dont le format est à votre choix.

Critères de correction

Classe Employe		5 points
Bon fonctionnement avec les matières exigées	7 x 1 point	7 points
Qualité de la programmation		3 points

Numéro B (20 points)

Matières étudiées : **polymorphisme, this, toString, equals, fichier de type texte, etc.**

Vous disposez du fichier de type texte nommé "**pays.txt**". Chaque ligne contient les informations d'un seul pays.

Le premier caractère représente le code d'un continent :

- '1' pour Afrique
- '2' pour Amérique
- '3' pour Asie
- '4' pour Océanie
- '5' pour Europe

Les autres sont : nom, capitale, superficie en km² et population.

Il y a au maximum **250** pays dans le fichier.

Réalisez un projet en Java, avec plusieurs classes, qui permet de réaliser les traitements ci-dessous :

1. Lire le fichier texte, remplir un tableau de pays, compter puis retourner le nombre effectif de pays lus.
2. Afficher seulement les 12 premiers pays lus (en utilisant entre autres, la redéfinition de **toString**) après la lecture.
3. Modifier le continent de la Chine, il doit devenir l'**Asie** ;
Modifier la capitale de la France, elle doit devenir **PARIS** ;
Changer la population de l'Allemagne : elle devient 10 fois la population lue ;
Augmenter de 4% la population de tous les pays d'**Afrique**.

Nous réaffichons les 15 premiers pays du tableau après ces modifications.
4. Afficher dans la console, les pays dont le nom est identique au nom de la capitale.
Exemple : **Luxembourg**

5. Déterminer puis afficher dans la console :
 - Le pays d'**Asie** dont la densité est la plus élevée.
 - Le pays d'**Amérique** dont la densité est la plus élevée.

NOTE : La densité représente le nombre d'habitants par km².
6. Déterminer puis afficher dans la console, le pays le plus peuplé
 - Du continent d'**Europe** ;
 - Du continent d'**Afrique**
7. Déterminer et afficher les informations dans la console :
 - a) Les pays dont le nom commence par une voyelle ;
 - b) Le pays d'**Océanie** dont la capitale contient le plus de lettres / caractères alphabétiques.
8. Trier (avec la méthode du tri rapide) selon les noms des pays, et afficher les 10 premiers pays après le tri.
9. Chercher (en utilisant la recherche dichotomique) les pays suivants :
 - **Canada, France, Japon, Mexique**
10. Créer le fichier texte avec le nom **Asie.txt** qui contient seulement les pays d'Asie.
Créer le fichier texte avec le nom **Europe.txt** qui contient seulement les pays d'Europe.

Critères de correction

Classe Nation		3 points
Bon fonctionnement avec les matières exigées	10 x 1.5 points	15 points
Qualité de la programmation		2 points

Bonne chance et bon succès !
Équipe du IFT 1170, Automne 2024