|  |  |  |
| --- | --- | --- |
| **IFT 1170** | **Énoncé du T.P. #1** | **Session été 2023** |

**Chargé de cours**: Michel Reid

***Préparation: chez-vous, à partir de la semaine du 1 mai***

***Réalisation:*** *durant vos démonstrations, les périodes de Pratique Libre (P.L.) à partir de la semaine du 1 mai.*

***Dépannage:*** *par votre démonstrateur (lors des séances de travaux pratiques).*

***Questions sur les travaux*** *: envoyez un courriel à l’adresse suivante :*  ***dift1170@iro.umontreal.ca***

***Date de remise:*** Au plus tard le*,* ***mardi 23 mai****.*

***Pénalité de retard:*** Chaque jour de retard entraînera une pénalité de 5 points (sur 25 points) par jour.

***Note***: Le travail en équipe de deux (au maximum) est permis. Vous ne remettez alors qu'un travail par équipe.

***Barème***: corrigé sur 30 points.

*Conseil amical:* N’attendez pas la semaine précédant la remise avant de commencer... vous n’aurez pas le temps!

***Matière exercée***: Éléments de base vus en Java, boîtes de dialogues de la classe **JOptionPane** et la POO à un niveau simple.

Assurez-vous d’avoir écrit vos coordonnées (nom et nom d’usager) ainsi que ceux de votre coéquipier (s’il y a lieu) en commentaires au début de chaque programme, une pénalité est prévue en cas d’oubli. Les équipes de 2 personnes ne doivent remettre qu’une seule copie du T.P.

Les équipes de 3 étudiants et plus sont interdites. Des pénalités sont à prévoir pour les équipes de 3 ou plus ayant clairement identifié tous les coéquipiers. Les équipes de 3 ou plus qui ne sont pas clairement identifiées comme tel et qui remettent plusieurs copies basées sur le même travail, feront l’objet d’un rapport au doyen selon les règles sur l’intégrité. Voir : http://www.integrite.umontreal.ca/

Le travail doit être remis **via Studium** par un des coéquipiers, l’autre effectuera une remise coéquipier**.**

**Pénalité si vous oubliez d’identifier l’un ou l’autre des coéquipiers. - 5 points**

**Numéro 1 (6 pts - Les classes Scanner, Integer et String)**

Réalisez un projet en Java qui contient une seule classe et seulement la méthode principale (main). Le projet permet :

1. De saisir une valeur entière au clavier (utilisez la méthode **nextInt** de la classe **Scanner**)
2. D’afficher la représentation binaire de cet entier, ainsi que le nombre de caractères nécessaires à cet affichage (voir exemple ci-dessous). N’utilisez que des appels à des méthodes des classes Integer et String pour obtenir ces deux informations. Par exemple :

valeur saisie 137

en binaire : 10001001

nombre de caractères : 8

valeur saisie 39

en binaire : 100111

nombre de caractères : 6

1. De recommencer avec un autre entier tant que l’usager le désire (utilisez la méthode next ou nextLine de la classe **Scanner** pour obtenir la réponse de l’usager)

**Critères de correction du numéro 1 du TP1:**

1. **Bon fonctionnement : 2 points**
2. **Utilisation des méthodes des classes Integer et String : 2 points**
3. **Présentation du programme et des résultats : 1 point**
4. **Commentaires : 1 point**

**Numéro 2 (5 pts - JOptionPane, Integer et String)**

Dans une nouvelle classe, avec uniquement la méthode main, refaites le numéro 1 de ce devoir en utilisant des boîtes de dialogues de la classe **JOptionPane** :

1. Utilisez la méthode showInputDialog pour saisir la valeur entière
2. Réutilisez le code du numéro 1 pour afficher la représentation binaire de cet entier, ainsi que le nombre de caractères nécessaires à cet affichage.
3. Utilisez la méthode showConfirmDialog pour demander si désire traiter un autre entier

**Critères de correction du numéro 2 du TP1:**

1. **Utilisation de showInputDialog : 2 points**
2. **Utilisation de showConfirmDialog : 2 points**
3. **Bon fonctionnement : 1 point**

**Numéro 3 (8 pts - la classe String : les chaînes de caractères)**

**Objectif : se familiariser au Java et à la POO**

Réalisez un projet en Java qui contient une seule classe, contenant seulement la méthode principale (main).

Le projet permet :

1. De déclarer et d’initialiser un tableau de chaînes de caractères représentant les noms et prénoms d’individus

Les chaînes de caractères respectent le format suivant :

a) Le(s) nom(s) et le(s) prénom(s) sont séparés par une virgule suivie d’une espace

b) S’il y a plus d’un nom (ou prénom), ils seront séparés par un trait d’union ou une espace

1. D’afficher, pour chacune des chaînes, le nombre de nom(s) et le nombre de prénom(s) qui la compose

Exemple d’exécution :

Pour la chaine "Lebeau Berube, Pierre Jean-Marc" :

Il y a 2 noms et 3 prenoms.

Pour la chaine "Lamoureux, Louis-Martin" :

Il y a 1 nom et 2 prenoms.

Voici un sous-ensemble du jeu de test qui sera utilisé pour la correction :

"Massie Chene, Ademar"

"Lacolle-Lepage, Pierre-Paul Andre"

"Desgagne-Lacourse, Fabien"

"Aucours, Yvon"

**Critères de correction du numéro 3 du TP1 :**

**1. Bon fonctionnement (faire afficher ce qui est demandé) : 5 points**

**(dont 1 point pour la gestion du singulier/pluriel)**

**2. Qualité de la programmation (on n’exige pas d’écrire des méthodes) : 2 points**

**3. Présentation du programme et des résultats** **: 1 point**

**Numéro 4 (11 pts - La POO, niveau très simple)**

**Objectif : se familiariser avec la notion d'encapsulation.**

Faisant suite au numéro 3, réalisez un projet en Java qui contient deux classes :

1. La classe **NomPrenom** qui comporte :

a) deux champs d’information : nom et prenom

b) un constructeur permettant de construire un objet **NomPrenom** à partir d’une chaîne de caractères (du même format qu’au numéro 3)

c) une méthode permettant de déterminer et de retourner le nombre de noms d’un **NomPrenom**

d) une méthode permettant de déterminer et de retourner le nombre de prénoms d’un **NomPrenom**

e) une méthode permettant d’afficher les noms et prénoms en inversant l’ordre des prénoms

Par exemple, pour "**Lacolle-Lepage, Pierre-Paul Andre**", l’affichage serait :

Lacolle-Lepage, Andre Paul Pierre

et pour "**Desgagne-Lacourse, Fabien**", l’affichage serait :

Desgagne-Lacourse, Fabien

f) d’autres méthodes que vous jugez utiles pour effectuer les tâches dans la classe **TestNomPrenom**

2. La classe **TestNomPrenom** :

a) dont la méthode principale (main) contient un tableau de 5 objets **NomPrenom** *construits* à partir des chaînes de caractères suivantes :

"Massie Chene, Ademar"

"Lacolle-Lepage, Pierre-Paul Andre"

"Desgagne-Lacourse, Fabien"

"Aucours, Yvon"

"Lacido, Remi Fassol"

b) qui contient des méthodes statiques (static) et leurs appels afin :

D’afficher les informations de chacun des 5 **NomPrenom** (en utilisant la méthode 1. e))

1. De compter et d’afficher le nombre total de prénoms pour tout le tableau
2. De compter et d’afficher le nombre total de noms pour tout le tableau

**Critères de correction du numéro 4 du TP1 :**

1. **Bon fonctionnement : 4 points**
2. **Qualité de la programmation de la classe NomPrenom : 3 points**
3. **Qualité de la programmation de la classe TestNomPrenom : 2 points**
4. **Présentation du programme et des résultats : 1 point**
5. **Commentaires : 1 point**

**La correction des travaux :**

La qualité de l'algorithme et de la programmation occupe une place importante dans l'évaluation d'un travail.

**Un programme qui fonctionne c’est bien, mais pas nécessairement suffisant pour mériter la meilleure note !**

Veuillez consulter les critères de correction de chacun des numéros du travail.

**Bon travail et bon succès dans vos études!**

**L’équipe du IFT 1170, automne 2023.**