Hiver 2018

Programmation avancée en C++

GIF-1003

Thierry EUDE, Ph.D

Travail individuel

à rendre avant
jeudi 1er mars 2018 14h
(Voir modalités de remise à la fin de l'énoncé)

Tout travail remis constitue une contribution originale et distincte de travaux remis par d'autres. Le plagiat est strictement défendu. Tout travail plagié obtiendra la note 0. De plus, les étudiants ou étudiantes ayant collaboré au plagiat seront soumis aux sanctions normalement prévues à cet effet par les règlements de l'Université.

Par ailleurs, vu l'importance des communications écrites dans le domaine de la programmation, il sera tenu compte autant de la présentation que de la qualité du français et ce, dans une limite de 10% des points accordés.

Travail Pratique #2
Développement de classes

UNIVERSITÉ LAVAL
Faculté des sciences et de génie
Département d'informatique
et de génie logiciel

Notre objectif final est de construire un outil de gestion de comptes des clients d'une banque. La série des travaux pratiques devrait tendre vers cet objectif. Chaque travail pratique constituera donc une des étapes de la construction de cet outil.

But du deuxième travail

- Apprivoiser le processus de développement d'une classe dans un environnement d'implémentation structuré.
- Respecter des normes de programmation et de documentation.
- Utiliser une classe dans un programme minimaliste

Classe Client

Cette classe permet de modéliser des clients d'une banque. Vous devrez implanter les spécifications qui suivent dans les fichiers `Client.h` et `Client.cpp`. La classe `util::Date` est disponible sur le site du cours.

Attributs de la classe Client

`m_noFolio`
Le numéro de folio du client. Doit être dans l'intervalle [1000, 10000[

`m_nom`
`m_prenom`
Des objets string. Le nom et le prénom du client. Doivent être dans un format valide tel que déterminé par la fonction `util ::validerFormatNom`. développée dans la première partie du projet (tp1).

`m_telephone`
Un objet string. Le numéro de telephone du client.
Le numéro de téléphone doit être dans un format valide tel que déterminé par la fonction `util::validerTelephone` développée dans la première partie du projet (tp1).

`m_dateOuverture`
La date d'ouverture d'un dossier client. Un classe Date (util::Date) vous est fournie, Ne couvre que l'intervalle [1970, 2037]. C'est la date courante au moment de l'inscription (en l'occurrence à la construction d'un client).

Méthodes de la classe

Constructeur de la classe. On construit un objet Client à partir de valeurs passées en paramètre.

```
Client(int p_noFolio, const std::string& p_nom, const std::string& p_prenom,
                  const std::string& p_telephone);
```

Méthodes d'accès aux attributs de la classe.
`reqNoFolio()`
`reqNom()`
`reqPrenom()`
`reqDateOuverture()`
`reqTelephone()`

Méthode qui permet d'assigner un nouveau numéro de téléphone au client courant. Le numéro de téléphone doit être dans un format valide selon la fonction `util::validerTelephone`.
`void asgTelephone(const std::string& p_telephone)`

`std::string reqClientFormate()`
Méthode qui retourne dans un objet `std::string` les informations correspondant à un client formatées sous le format suivant :

```
Client no de folio : 5000
Joe Blo
418 656-2131
Date d'ouverture : Lundi le 12 fevrier 2018
```
Utilisez la classe ostringstream du standard pour formater les informations sur le client comme demandé.

Opérateur de comparaison d'égalité. La comparaison se fait sur la base de tous les attributs.
`bool operator==(const Client& p_client)`

`bool operator<(const Client& p_client)`
Opérateur de comparaison d'infériorité. La comparaison se fait sur la base du numéro de client.

Fonction de validation

Il s'agit de compléter le module de fonctions de validation développé dans le TP1 dans le cas où des erreurs vous auraient été signalées. Un testeur est à votre disposition sur la page de ce travail pratique pour vous aider à le faire le cas échéant.
Vous devez insérer l'ensemble des fonctions dans le namespace **util**.

Documentation

La classe `Client` ainsi que toutes les méthodes devront être correctement documentées pour pouvoir générer une documentation complète à l'aide de l'extracteur DOXYGEN. Des précisions sont fournies sur le site Web pour vous permettre de l'utiliser (syntaxe et balises à respecter etc.).
Vous devez respecter les normes de programmation adoptée pour le cours. Ces normes de programmation sont à votre disposition dans la section "Notes de cours / Présentations utilisées en cours / Normes de programmation en C++ " du site Web du cours.
Aussi, comme indiqué dans ces normes, vous devez définir un espace de nom (namespace). Il devra porter le nom suivant : `banque` (en minuscules).

Utilisation

Après avoir implanté la classe demandée, vous devez écrire un programme interactif. Le but est simple, il s'agit simplement d'obtenir interactivement avec l'usager, les données nécessaires pour créer un "`Client`", puis de le modifier.
Rappelons qu'un objet *`Client`* ne peut être construit qu'avec des valeurs valides. C'est la responsabilité du programme principal qui l'utilise de s'assurer que ces valeurs sont valides.
Les critères de validité ont été énoncés dans la description des attributs de la classe et dans l'énoncé des fonctions de validation.
Une fois toutes les données validées, vous créez un objet *`Client`* et vous demandez à l'objet créé de retourner une chaîne formatée (*`reqClientFormate()`*) pour l'afficher dans la console. Vous demandez ensuite à l'usager un nouveau numéro de téléphone pour l'assigner au client. Il s'agit ensuite de prévoir un affichage permettant de constater que la modification a bien été effective. Et c'est tout pour l'instant…
Voir l'exemple d'exécution fourni à la fin de cet énoncé. Cependant, respectez strictement l'ordre et les données saisies (pensez au correcteur qui va devoir évaluer votre travail).

Modalités de remise, bien livrable

Le deuxième travail pratique pour le cours GIF-1003 Programmation avancée en C++ est un travail individuel. Vous devez remettre le code source dans un espace de travail (workspace) Eclipse mis dans une archive **7z** (voir le tutoriel sur la page des travaux pratiques dans contenu et activités), en utilisant le dépôt de l'ENA :
https://sitescours.monportail.ulaval.ca/ena/site/evaluation?idSite=86617&idEvaluation=337828&onglet=boiteDepots
Ce travail est intitulé TP2. Aucune remise par courriel n'est acceptée.

⚠ Très important ⚠

Attention, vérifiez qu'une fois déplacé et décompressé, votre workspace est toujours fonctionnel (il doit donc être « portable »). Pensez au correcteur ! Sachez qu'il utilisera la machine virtuelle fournie pour le cours.
L'ENA ne permet pas de retourner les TP corrigés comme évaluation sommative, les retours seront faits dans évaluation formative (retour correction TP2), mais les notes sont versées dans l'évaluation sommative.

Date de remise

Ce travail doit être rendu avant le jeudi 1er mars 2018 à 14h00. Pour tout retard non motivé (voir plan de cours; motifs acceptables pour s'absenter à un examen), la note 0 sera attribuée.

Critères d'évaluation

1) Respect des biens livrables
2) Respect des normes de programmation, et de documentation
3) Structures, organisation du code (très important!)
4) Exactitude du code
5) Utilisation des classes, i.e. le programme principal

Particularités du barème

- *Si des pénalités sont appliquées, elles le sont sur l'ensemble des points.*
- *Si un travail comporte ne serait-ce qu'une erreur de compilation, il sera fortement pénalisé, et peut même se voir attribuer la note zéro systématiquement.*
- *Il est très important que votre travail respecte strictement les consignes indiquées dans l'énoncé, en particulier les noms des méthodes, les noms des fichiers et la structure de développement sous Eclipse sous peine de fortes pénalités*

Bon travail

Exemple d'exécution

```
Bienvenue a l'outil client
---------------------------------------
Entrez le numero de folio du client [1000, 10000[ :
12345
Le numero de folio n'est pas entre 1000 et 10000 exclusivement

Entrez le numero de folio du client [1000, 10000[ :
5678

Entrez le prénom :
J9oe
Le prénom n’est pas valide

Entrez le prénom :
Joe

Entrez le nom :
Bl;o
Le nom n’est pas valide

Entrez le nom :
Blo

Entrez le numero de telephone XXX CCC-CCCC:
123 999.9999
Le numero de telephone n'est pas valide.

Entrez le numero de telephone XXX CCC-CCCC:
418 656-2131

Fiche client
-----------------------------------------------
Client no :5678
Joe Blo
418 656-2131
Date d'ouverture : Lundi le 12 fevrier 2018

Entrez un nouveau numero de telephone XXX CCC-CCCC:
418 656-5669

Fiche client
-----------------------------------------------
Client no :5678
Joe Blo
418 656-5669
Date d'ouverture : Lundi le 12 fevrier 2018

Fin du programme!
```