

fl (<https://profile.intra.42.fr/searches>)

(<https://profile.intra.42.fr>)

ÉCHELLE DU PROJET CPP MODULE 05

(/PROJETS/CPP-MODULE-05)

Vous devez évaluer 1 étudiant dans cette équipe

Dépôt Git

`git@vogsphere.msk.21-school.ru:vogsphere/intra-uuid-193aaa9`

Introduction

Veillez respecter les règles suivantes :

- Restez poli, courtois, respectueux et constructif tout au long du processus d'évaluation. Le bien-être de la communauté en dépend.
- Identifier avec l'étudiant ou le groupe dont le travail est évalué les dysfonctionnements possibles de son projet. Prendre le temps de discuter et de débattre des problèmes éventuellement identifiés.
- Il est important de prendre en compte les différences de compréhension entre vos pairs quant aux instructions du projet et à l'étendue de ses fonctionnalités. Gardez toujours l'esprit ouvert et notez-les aussi honnêtement que possible. La pédagogie n'est utile que si l'évaluation par les pairs est menée avec sérieux.

Lignes directrices

- Notez uniquement le travail qui a été rendu dans le dépôt Git de l'étudiant ou du groupe évalué.
- Vérifiez que le dépôt Git appartient bien à l'étudiant(e). Assurez-vous que le projet est bien celui attendu. Vérifiez également que « git clone » est utilisé dans un dossier vide.
- Vérifiez soigneusement qu'aucun alias malveillant n'a été utilisé pour vous tromper et vous faire évaluer quelque chose qui n'est pas le contenu du dépôt officiel.
- Pour éviter toute surprise et le cas échéant, revoir ensemble les scripts utilisés

pour faciliter la notation (scripts de test ou d'automatisation).

- Si vous n'avez pas terminé le devoir que vous allez évaluer, vous devez lire l'intégralité du sujet avant de commencer le processus d'évaluation.

- Utilisez les indicateurs disponibles pour signaler un référentiel vide, un programme non fonctionnel, une erreur de norme, une tricherie, etc.

Dans ces cas, le processus d'évaluation est terminé et la note finale est de 0, ou -42 en cas de tricherie. Cependant, sauf en cas de tricherie, il est fortement recommandé aux étudiants de relire ensemble les travaux rendus afin d'identifier les erreurs à éviter.

- Vous ne devriez jamais avoir à modifier un fichier, sauf le fichier de configuration, s'il existe.

Si vous souhaitez modifier un fichier, prenez le temps d'en expliquer les raisons à l'étudiant évalué et assurez-vous que vous êtes tous les deux d'accord.

- Il est également nécessaire de vérifier l'absence de fuites mémoire. Toute mémoire allouée sur le tas doit être correctement libérée avant la fin de l'exécution.

Vous êtes autorisé à utiliser tous les outils disponibles sur votre ordinateur, tels que leaks, valgrind ou e_fence. En cas de fuite de mémoire, cochez l'option appropriée.

Pièces jointes

sujet.pdf (<https://cdn.intra.42.fr/pdf/pdf/40215/sujet.pdf>)

Tests préliminaires

En cas de suspicion de triche, l'évaluation s'arrête ici. Utilisez l'indicateur « Triche » pour le signaler. Prenez cette décision avec calme et prudence, et utilisez ce bouton avec prudence.

Prérequis

Le code doit être compilé avec C++ et les options -Wall -Wextra -Werror. N'oubliez pas que ce projet doit respecter la norme C++98. Par conséquent, les fonctions ou conteneurs C++11 (et versions ultérieures) ne sont PAS attendus.

L'un de ces éléments signifie que vous ne devez pas noter l'exercice en question :

- Une fonction est implémentée dans un fichier d'en-tête (à l'exception des fonctions de modèle).

- Un Makefile se compile sans les drapeaux requis et/ou un autre compilateur que c++.

Chacun de ces éléments signifie que vous devez marquer le projet avec « Interdit ».

Fonction":

- Utilisation d'une fonction "C" (*alloc, *printf, free).

- Utilisation d'une fonction non autorisée dans les consignes d'exercice.

- Utilisation de « using namespace » ou le mot-clé « ami ».
- Utilisation d'une bibliothèque externe ou de fonctionnalités de versions autres que C++98.

Oui

Non

Ex00 : Maman, quand je serai grand, je veux être bureaucrate !

Comme d'habitude, la fonction principale doit contenir suffisamment de tests pour prouver que le programme fonctionne comme prévu. Dans le cas contraire, ne notez pas cet exercice. Si une classe non-interface n'est pas conforme à la forme canonique orthodoxe, ne notez pas cet exercice.

Bureaucrate

Il existe un Makefile qui se compile en utilisant les indicateurs appropriés.

Il existe une classe de bureaucrates. Son nom est constant.

Sa note varie de 1 (la plus élevée) à 150 (la plus basse).

Des exceptions sont levées lorsque vous essayez de créer un bureaucrate avec une note trop élevée ou trop basse.

Il existe des accesseurs pour les attributs.

Il existe des fonctions permettant d'incrémenter ou de décrémenter la note.

Elles génèrent des exceptions lorsque cela est approprié. N'oubliez pas qu'incrémenter une note de 3 donne la note de 2 (1 étant la note la plus élevée).

Les exceptions utilisées héritent de `std::exception`, ou de quelque chose dérivé de `std::exception` (c'est-à-dire qu'elles sont capturables comme `std::exception & e`).

Il existe un opérateur << pour la surcharge ostream qui renvoie les informations du Bureaucrate.

Oui

Non

Ex01 : Formez-vous, asticots !

Comme d'habitude, la fonction principale doit contenir suffisamment de tests pour prouver que le programme fonctionne comme prévu. Dans le cas contraire, ne notez pas cet exercice. Si une classe non-interface n'est pas conforme à la forme canonique orthodoxe, ne notez pas cet exercice.

Formulaire

Il existe un Makefile qui se compile en utilisant les indicateurs appropriés.

Il existe une classe Form.

Il a un nom, un booléen qui indique s'il est signé (au début ce n'est pas le cas), une note requise pour le signer et une note requise pour l'exécuter.

Les noms et les grades sont constants.

Tous ces attributs sont privés et non protégés.

Les notes des formulaires suivent les mêmes contraintes que celles du Bureaucrate
(exceptions, 1 = le plus élevé, 150 = le plus bas, et ainsi de suite).

Il existe des accesseurs pour les attributs et un opérateur << pour la surcharge
ostream qui affiche l'état complet du formulaire.

Il existe une fonction membre `Form::beSigned()` qui fonctionne comme décrit par le sujet.

Il existe une fonction `Bureaucrat::signForm()` qui fonctionne comme décrit par le sujet.

Oui

Non

Ex02 : Non, vous avez besoin du formulaire 28B, pas du 28C...

Comme d'habitude, la fonction principale doit contenir suffisamment de tests pour prouver que le programme fonctionne comme prévu. Dans le cas contraire, ne notez pas cet exercice. Si une classe non-interface n'est pas conforme à la forme canonique orthodoxe, ne notez pas cet exercice.

Des formulaires qui font réellement quelque chose

Il existe un Makefile qui se compile en utilisant les indicateurs appropriés.

Il existe des formes concrètes qui sont conformes aux spécifications de la matière (notes
requis, noms et actions).

Ils ne prennent qu'un seul paramètre dans leur constructeur, qui est leur cible.

Il existe une méthode `Form::execute(Bureaucrat const & executor)` qui fonctionne comme
spécifié par le sujet.

Soit cette méthode est pure et les vérifications de qualité sont implémentées dans chaque
sous-classe, soit cette méthode effectue les vérifications, puis appelle une autre méthode
dans la classe dérivée qui exécute uniquement l'action.

Ces deux techniques sont valables.

Il existe un `Bureaucrat::executeForm(Form const & form)` qui fonctionne comme spécifié
par le sujet.

Oui

Non

Ex03 : Au moins, c'est mieux que de faire du café

Comme d'habitude, la fonction principale doit contenir suffisamment de tests pour prouver que le programme fonctionne comme prévu. Dans le cas contraire, ne notez pas cet exercice. Si une classe non-interface n'est pas conforme à la forme canonique orthodoxe, ne notez pas cet exercice.

Interne

Il existe un Makefile qui se compile en utilisant les indicateurs appropriés.

Il y a une classe de stagiaires.

Il dispose d'une fonction makeForm() qui fonctionne comme spécifié par le sujet.

Oui

Non

Bon envoi

La fonction makeForm() doit utiliser une sorte de tableau de pointeurs vers des fonctions membres pour gérer la création de formulaires.

S'il utilise une méthode impure, comme les branches if/elseif/elseif/else, ou d'autres trucs laids comme ça, veuillez considérer cela comme faux.

Oui

Non

Notes

N'oubliez pas de cocher le drapeau correspondant à la défense

D'accord

Projet exceptionnel

Travail vide

Aucun fichier auteur W

Compilation invalide

Norme

Cheat d Crash

Fuites

I Fonction interdite

Conclusion

Laissez un commentaire sur cette évaluation

Гр о вер као ф лайн

МДК. Пир р ас с казал

Terminer l'évaluation