

Implémentation des piles et files

Le but de ce laboratoire est l'implantation de l'interface d'une pile et d'une file génériques. Parmi les exemples vus en classe, nous vous avons fourni des implantations d'une pile dans un tableau dynamique et d'une file dans une liste simplement chaînée. Il est question ici de refaire l'implantation d'une pile et d'une file dans une liste simplement chaînée et dans un tableau dynamique respectivement.

Documentations : voir le site Web du cours. Vous y trouverez la description des commentaires attendus dans un programme ainsi que des normes de programmation en vigueur dans notre cours.

Travail à faire

Nous vous fournissons 2 paquetages:

- Pile, implémentation dans une liste simplement chaînée
- File, implémentation dans un tableau dynamique

Dans chacun d'eux, vous devez compléter le fichier .hpp étant donné le modèle d'implémentation décrit dans la partie privée de la classe décrite dans chaque paquetage.

Important

1. Nous vous fournissons des tests unitaires Google Test que vous pouvez utiliser pour vérifier votre implémentation.
2. Vous êtes tenu de faire la gestion des exceptions dans les méthodes que vous avez à implémenter. Référez-vous à la documentation Doxygen fournie avec l'énoncé pour connaître les types d'exception que vous avez à gérer pour chacune d'elles. Vous devez utiliser le cadre de la théorie du contrat que nous avons préparé dans les fichiers ContratException.cpp et ContratException.h disponible sur le site du cours.
3. Vous devez documenter chaque méthode, que vous avez à implémenter, avec les commentaires de Doxygen. Référez-vous à section de Doxygen sur le site Web du cours ainsi qu'aux exemples de cette semaine pour découvrir les commentaires que vous avez à écrire. Assurez-vous de respecter les normes de programmation en vigueur disponible également sur le site du cours.
4. Vous devez générer la documentation par Doxygen pour chacune des 2 implémentations demandées. À cette fin, nous vous fournissons un fichier de configuration IFT2008.doxyfile que vous pouvez utiliser tel quel.
5. Nous vous encourageons à utiliser au maximum la partie privée de la classe afin d'y mettre des fonctions utilitaires privées. Elles permettent d'augmenter la lisibilité du code et de réduire la duplication de code identique.