

Travail dirigé : Implémentation de Liste

Objectif:

Le but de ce laboratoire est de vous faire découvrir un autre modèle de base pour implémenter l'interface d'une liste ordonnée. Nous avons vu en classe une implémentation possible d'une liste doublement chaînée. Il est question ici de refaire l'implémentation de quelques opérateurs de la liste dans un autre modèle utilisant le chaînage ainsi que la gestion dynamique de la mémoire. est d'implémenter .

Travail à faire

Vous devez implémenter une liste ordonnée utilisant un chaînage simple circulaire (anneau). Nous vous fournissons une interface à respecter dans le fichier `Liste.h` , et vous devez compléter le fichier `Liste.hpp` .

Complément

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` disponibles 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 disponibles également sur le site du cours.
4. Vous devez générer la documentation en format HTML. À cette fin, nous vous fournissons un fichier de configuration `sdd.doxyfile` que vous pouvez utiliser tel quel.
5. Nous vous encourageons à utiliser au maximum la partie privée de la classe afin d'y ajouter des fonctions utilitaires privées. Elles permettent d'augmenter la lisibilité du code et de réduire la duplication de code identique.