M3103 – Algorithmique avancée

TP 3.2 *(associé au Cour 3)*

Autour de la STL – ***stack, queue***

(1 séance encadrée, une séance en AA [en S4])

`stack` (séance encadrée **tp321**)

Pour cette séance :

- créer un projet **M3103 - MyStack** dans NetBeans (cf. Aide NetBeans).
- copier les fichiers à compléter disponibles dans /users/info/pub/2a/M3103/**tp321** dans le répertoire créé par NetBeans (cf. Aide NetBeans).
- importer les fichiers dont l'extension est `.h` ou `.cpp` (cf. Aide NetBeans).

Excercie 1 : Premier contact avec la classe stack

Créer une configuration **PremierContact** dans le projet **M3103 - MyStack** avec comme point d'entrée le fichier **`premiercontact.cpp`**.

Compléter le fichier **`premiercontact.cpp`** afin de produire une trace d'exécution la plus fidèle possible de celle qui est proposée.

Excercie 2 : Affichage itératif de droite à gauche d'une liste

Dans le TP2, vous avez implanté la classe **PileListeChainee** (pile par une liste chainée de cellules) et utilisé cette classe pour implanter la méthode **`afficheDGIter()`**.

Dans la Classe **ListeChainee** ajouter, implanter et tester une méthode **`afficheDGIterWithStack()`** qui utilise une **`stack`**.

Excercie 3 : Suis-je bien parenthésée ?

Créer une configuration **BienParenthesee** dans le projet **M3103 - MyStack** avec comme point d'entrée le fichier **`bienparenthesee.cpp`**.

On veut écrire le prédicat qui vérifie qu'une séquence de caractères représentée sous forme d'un **`string`** dans laquelle les caractères `'{'`, `'}'`, `'('`, `')'`, `'['` et `']'` servent de parenthèses ouvrantes (`'{'`, `'('`, `'['`) et fermantes (`'}'`, `')'`, `']'`) est bien parenthésée.

Une expression est dite est *bien parenthésée* si :

- Pour toute parenthèse fermante d'un certain type, il existe une parenthèse ouvrante du même type qui la précède, et qui n'a pas encore été fermée.
- Toute parenthèse ouvrante d'un certain type est fermée par une parenthèse fermante du même type.

Les expressions suivantes sont bien parenthésées :

- ([3+5]∗(2∗9))
- 1+4
- ([(2)])

Les expressions suivantes ne sont pas bien parenthésées :

- (3 + 5] : les types de parenthèses ne correspondent pas.
- ((3) : une des deux parenthèses n'a pas été fermée.
- (4)) : on ferme une parenthèse qui n'a pas été ouverte.

Dans un premier temps, résoudre ce problème (algorithme et implantation) avec un seul type de parenthèse ouvrante et fermante (`estBienParentheseeSimple(string maChaine)`).

Résoudre ensuite ce problème (algorithme et implantation) avec les trois types de parenthèses ouvrantes et fermantes (`estBienParenthesee(string maChaine)`).

queue (séance AA **tp322**)

Pour cette séance :

- créer un projet **M3103 - MyQueue** dans NetBeans (cf. Aide NetBeans).
- copier les fichiers à compléter disponibles dans /users/info/pub/2a/M3103/**tp322** dans le répertoire créé par NetBeans (cf. Aide NetBeans).
- importer les fichiers dont l'extension est `.h` ou `.cpp` (cf. Aide NetBeans).

Excercie 4 : Premier contact avec la classe queue

Créer une configuration **PremierContact** dans le projet **M3103 - MyQueue** avec comme point d'entrée le fichier `premiercontact.cpp`.

Compléter le fichier `premiercontact.cpp` afin de produire une trace d'exécution la plus fidèle possible de celle qui est proposée.

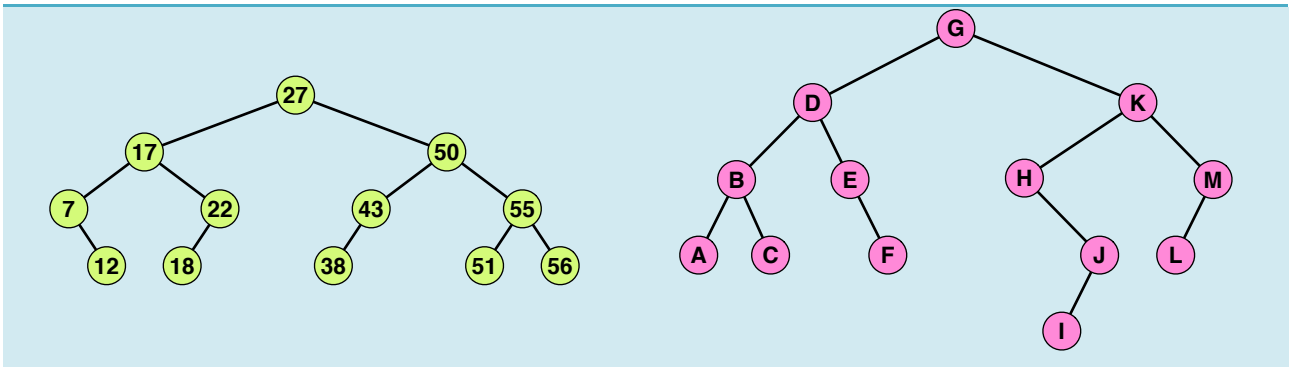
Excercie 5 : Affichage en largeur d'abord d'un arbre

Dans le TP 4 avez implanté un arbre binaire de recherche (classe **ArbreNoeudBinaireRecherche**) avec des nœuds binaires (classe **NoeudBinaire**) avec, en particulier, trois méthodes d'affichage en profondeur d'abord (`affichePrefixe()`, `afficheInfixe()`, `affichePostFixe()`).

Dans l'affichage en largeur d'abord d'un arbre (binaire ou n-aire), on affiche les valeurs contenues dans l'arbre en faisant un parcours par niveau.

Exemples :

Arbre / Trace du parcours en largeur d'abord



27, 17, 50, 7, 22, 43, 55, 12, 18, 38, 51, 56

G, D, K, B, E, H, M, A, C, F, J, L, I

Pour implanter cet algorithme, on a besoin d'une file d'attente.

Dans la Classe **ArbreNoeudBinaireRecherche** ajouter, implanter et tester une méthode **`afficheLargeur()`** qui utilise une **`queue`**.