

Exercice 1 On considère des enfants qui vont jouer en boucle autour d'un toboggan.

1. Écrivez une classe **Enfant** contenant un nom et un nombre figurant le nombre de fois où ils ont utilisé un toboggan, ainsi qu'un constructeur.
2. Déclarez une classe **Cellule**, doublement chaînée et destinée à recevoir des enfants, ainsi qu'un constructeur prenant en argument un **Enfant**, et qui initialise les champs **precedent** et **suivant** avec **this**.

Exercice 2 : File.

On va créer une classe **FileToboggan**, qui représente une file d'enfants qui veulent glisser sur un toboggan. Après avoir glissé, un enfant retourne faire la queue. Les enfants sont modélisés par une liste doublement chaînée circulaire (voir Fig.1).

1. Créer une telle classe **FileToboggan**, qui contient un attribut de type **Cellule courant** qui représente la cellule de l'enfant qui est en train de glisser. Ajouter un attribut **int nbMax** qui correspond au nombre maximal de fois qu'un enfant peut glisser.
2. Écrire un constructeur qui ne prend que le nombre maximal de glissades possibles, et initialise **courant** à **null**.
3. Écrire une méthode **ajouter(String nom, int nb)** qui ajoute un enfant de nom **nom** qui a glissé **nb** fois à la file d'enfants en queue pour glisser. Attention à bien gérer le cas où la file est vide.
4. Ajouter une méthode **supprimer(String nom)** qui supprime de la file l'enfant de nom **nom**. On suppose qu'il y a au plus un tel enfant.
5. Ajouter une méthode **affiche()** qui affiche **courant** et tous les enfants en queue pour glisser.
6. Ajouter une méthode **Enfant unTour()** qui fait glisser **courant** en affichant le nom de l'enfant qui glisse. Si son nombre **nb** de fois qu'il a glissé est (strictement) plus petit que **nbMax**, on le rajoute à la file. Sinon il doit laisser la place aux autres (et donc le retire de la liste). Si un enfant est retiré de la file, la méthode retourne cet **Enfant**, et sinon elle retourne **null**.

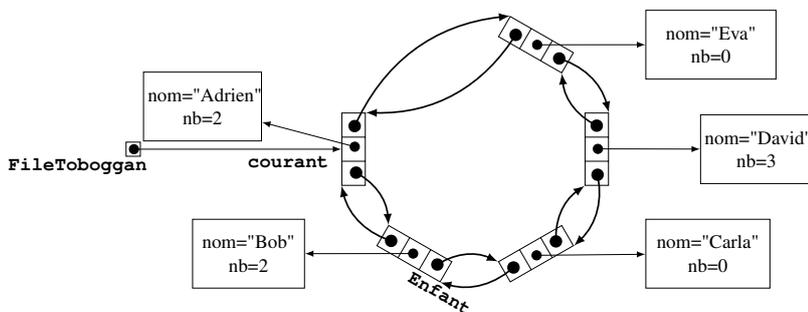


Fig.1

Par exemple, dans la situation de la Fig.1 un appel à **ajouter("Fred", 0)** mène à

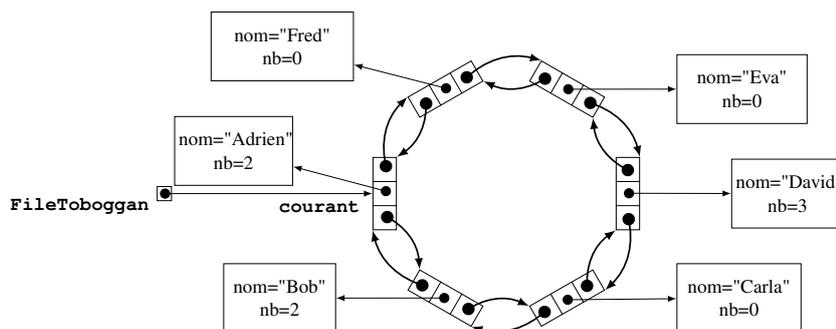


Fig.2

Dans la situation de la Fig.2 l'enfant de nom Fred est le dernier enfant qui glissera et la prochaine (après que l'enfant de nom Adrien aura glissé) sera l'enfant de nom Bob (si on lit dans le sens antihoraire).

Exercice 3 : Toboggan.

1. Écrire une classe **Toboggan**, qui contient un attribut de type **FileToboggan file**, avec un constructeur qui permet d'initialiser cet attribut.
2. Ajouter une méthode **int jouer()** qui fait glisser les enfants jusqu'à ce que **file** soit vide et renvoie le nombre des tours effectués. On pourra utiliser la méthode **unTour()** de l'exercice précédent.
3. Modifiez la méthode précédente pour qu'elle retourne un objet de type **ArrayList<Enfant>** qui contient les enfants sortis de la file, dans l'ordre chronologique de leur sortie (on utilisera le fait que la méthode **unTour()** retourne, le cas échéant, l'enfant qui sort).

Exercice 4 : Préséance. (Facultatif)

Dans la classe **FileToboggan** :

1. Ajouter une méthode **void laissePasser()** dans laquelle l'enfant **courant** échange sa position avec l'**Enfant** sensé glisser après lui si celui-ci a glissé strictement moins de fois.
2. Ajouter une méthode **Enfant unTourPre()** dans laquelle l'enfant **courant** glisse, sauf si l'**Enfant** après lui a glissé strictement moins de fois, auquel cas il décide de le laisser glisser avant lui.
3. Modifier la méthode précédente pour gérer le cas d'égalité de la façon suivante : si **courant** et l'enfant qui doit glisser après lui ont glissé le même nombre de fois, alors avec probabilité $1/3$ c'est **courant** qui glisse, avec probabilité $1/3$ il cède sa place à l'autre enfant et glissera après lui, et avec probabilité $1/3$ l'enfant **courant** décide qu'il a assez glissé auquel cas il glisse une dernière fois puis quitte la file.