Conception

# **Une classe : Données** Contient les deux structures de données suivantes :

* La liste des noeuds (documents).
* La liste des arcs.

Ces structures de données sont créées au fur et à mesure par l’intermédiaire d’une méthode update(Log l), qui récupère les informations désirées dans les attributs du Log.

Deux procédures (amies ?)/méthodes (?) font partie de Données (?) :

* TopTen() parcourt la liste des noeuds, créée la liste des noeuds inversée, et affiche les 10 premiers.
* Graphe() parcourt la liste des noeuds puis la liste des arcs pour créer un fichier .dot correspondant aux structures de données.

**Une classe : Fichier**

Constructeur -> on lui passe en paramètre le nom du fichier .log qu’il lit ligne par ligne par l’appel à la méthode getLine().

**Une classe : Logs**

Constructeur -> on lui passe en paramètre une ligne du fichier anonyme.log via la méthode getLine() de Fichier et trie les informations dans des attributs privés :

* ip
* url\_doc
* url\_referer
* quant\_data
* status
* heure (date ?)
* action
* navigateur

La classe a une fonction publique (“getteur”) pour chaque attribut privé. Elle sert à transmettre les bonnes informations au bon moment.

**Main :**

* Récupère les options (-t, -g, -e)
* Gère les cas particuliers (cas limites et cas d’erreurs)
* Instancie un Fichier, un Log et un Donnees
* Fait appel à getLine, fournit la ligne à Log, puis fournit le Log à Données
* Est-ce que le main décide de fournir le Log à Données que si celui-ci est “valide” ? -> en fonction du status, de l’action, des options ?
* Fait appel à TopTen(), et en fonction de l’option -g, fait appel à Graphe()