**R308 - consolidation de la programmation : TP3**
24 septembre 2022

# Gestion d'un parking

Une société de gestion de parkings de voitures souhaite mettre en place un système automatique d'affichage en temps réel de nombre de places disponibles dans un parking.

Un parking est caractérisé par : son adresse, sa capacité d'accueil, et le nombre de portails. Chaque portail est doté d'un écran affichant un message précisant les informations suivantes :

— le numéro du portail,

— le nombre de places libres dans le parking (ou si le parking est complet),

— le nombre d'entrées et le nombre de sorties effectuées par le portail.

Donner les classes nécessaires pour effectuer cette simulation.

# L'Algèbre d'Allen

Un intervalle $X$ est définie par deux bornes : $d, f \in \mathbb{R}$ avec $d \leq f$. L'algèbre d'Allen est une algèbre définie sur les intervalles. Elle comporte 13 relations. Ces relations sont illustrées à la figure 1.

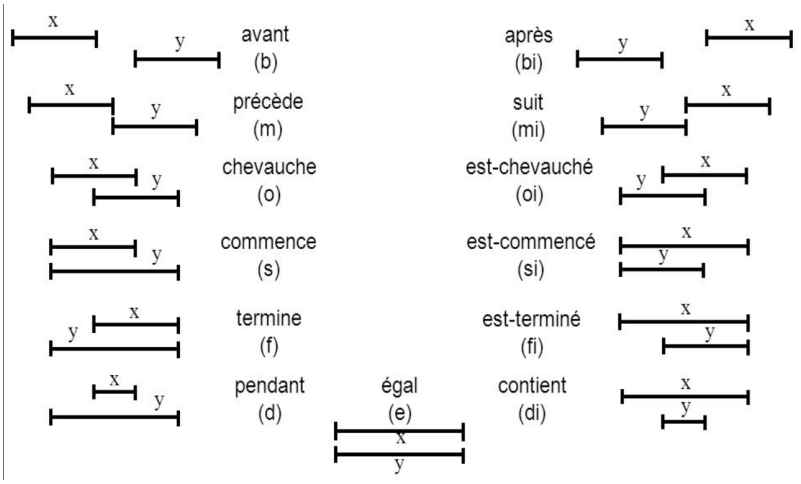


FIGURE 1 – Relations d'Allen définies sur les intervalles $X$ et $Y$

Développer une classe Python nommée `Intervalle`, qui permet de representer des intervalles et offrir l'ensemble des relations définies dans l'algèbre d'Allen. Sur la figure 1, le nom de la méthode qui implémente une relation est donné entre parenthèses. Soient $X, Y$ deux objets valides instanciés à partir

de la classe `Intervalle`. L'instruction `X.b(Y)` teste alors si l'intervalle `X` est avant l'intervalle `Y`. L'instruction `X.fi(Y)` teste si l'intervalle `X` est terminé par l'intervalle `Y`.

1. Donner la méthode constructeur de la classe `Intervalle`. On rappelle que la méthode constructeur doit seulement construire d'objets valides (conformes à la définition d'un intervalle)
2. Définir la méthode `e()`qui implémente la relation `égale`
3. Donner les codes des méthodes : `b(), m(), o(), s(), f()` et `d()`
4. La relation `après` (notée `bi`) est l'inverse de la relation `avant` (notée `b`) : (`X`est après `Y` ) $\Longleftrightarrow$ (`Y` est avant `X`). En profitant de cette observation proposer une implémentation de la méthode `bi()` en utilisant la méthode `b()`.

5. Donner un programme qui permet de tester le bon fonctionnement de la classe `Intervalle` que vous proposez.

## Python et les atomes

On veut développer une classe python nommée `Atom` qui permet de representer un atome d'un élément chimique. On commence par donner quelques rappels sur la structure atomique des éléments chimiques.
yellow

- L'atome d'un élément chimique est composé d'un *noyau* autour duquel tournent des *électrons*.
- Un électron a une charge électrique élémentaire négative et une masse négligeable.
- Le noyau est composé de *nucléons*. Un nucléon peut être de deux types :
  - *Neutron* : une particule de charge électrique neutre et dont la masse est : $1,675 \times 10^{-27}$ kg
  - *Proton* : une particule de charge électrique élémentaire positive et dont la masse est : $1,673 \times 10^{-27}$ kg
- La charge totale d'un atome est neutre comme il comporte autant d'électrons que de protons.
- Les électrons sont disposés en couches autour du noyau. Chaque couche a une capacité maximale de **8 électrons** sauf la première couche dont la capacité est limitée à **2 électrons**.
- Un électron ne peut se positionner sur une couche si la couche précédente n'est pas totalement remplie.

- Le nombre d'électrons sur la couche externe détermine les caractéristiques chimiques de l'élément. Deux atomes différents ayant le même nombre d'électrons sur leurs couches extérieures sont dits de la même famille.
- Deux atomes qui ont le même nombre de protons (ou d'électrons) mais un nombre différent de neutrons sont dits **isotopes**.
- Un élément chimique est défini par : ${}^{A}_{Z}X$ où $X$ est le symbole de l'élément (ex. H pour Hydrogène, C pour Carbone, ...), $A$ est le nombre de nucléons (dits aussi nombre de masse) et $Z$ est le nombre atomique qui représente le nombre de charges (nombre de protons ou d'électrons)

**Exemple :** Le carbone est donné par ${}^{12}_{6}C$. L'atome de carbone comporte alors $12 - 6 = 6$ neutrons. Il a $6 - 2 = 4$ électrons sur sa couche externe. Sa masse est de : $(6 \times 1,675 \times 10^{-27}) + (6 \times 1,673 \times 10^{-27})$. Le carbone ${}^{14}_{6}C$ est un isotope du carbone ${}^{12}_{6}C$.

La classe `Atom` doit permettre d'instancier des objets qui modélisent des éléments chimiques. Pour chaque élément on doit être capable de calculer sa masse totale, le nombre d'électrons sur la couche extérieure et de le comparer avec un autre élément.

1. Donner l'ensemble des attributs de la classe `Atom` en précisant pour chaque attribut les informations suivantes :
   - Nature : attribut d'instance ou attribut de classe.
   - Type : domaine de valeurs.
   - Contraintes à respecter sur les valeurs de l'attribut.
2. Donner le code de la méthode constructeur de la classe `Atom`.
3. Donner le code de la méthode `masse()` qui retourne la masse totale de l'élément
4. Donner le code de la méthode `NbElecExt()` qui retourne le nombre d'électrons sur la couche externe.
5. Donner le code de la méthode `memeFamille(elem)` qui teste si l'élément est de la même famille d'un autre élément *elem*.
6. Donner le code de la méthode `isotope(elem)` qui teste si l'élément est un isotope de l'élément *elem*.