



PROGRAMMATION ORIENTÉE OBJET

Informatique de gestion - bloc 2

Haute-École de Namur-Liège-Luxembourg

---

## Laboratoire 4 : exercice récapitulatif

---

### Objectifs

- Revoir les notions vues dans les premiers laboratoires

Un centre de compétences sponsorisé par la Ville de Namur et la Région Wallonne organise des formations auxquelles peuvent assister diverses personnes (enseignants, chômeurs et autres). Toutes les formations ont un nombre maximum de places mais ce nombre ne dépassera pas 25.

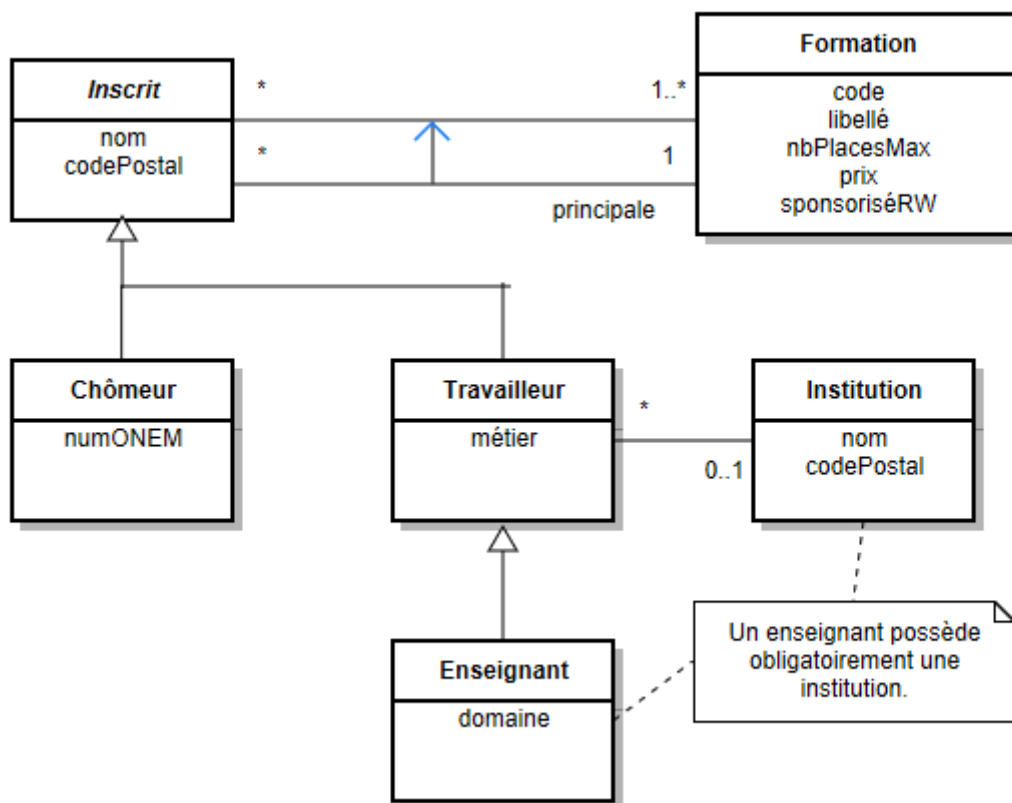
Tout inscrit donne son nom et le code postal de son domicile. Le chômeur donne aussi le numéro sous lequel il est inscrit à l'ONEM. Un travailleur est aussi caractérisé par son métier ; un enseignant possède un domaine de compétences et appartient obligatoirement à une institution (unique dans notre cas). Le travailleur peut être rattaché à une institution (entreprise ou autre) dont on donne le nom et le code postal.

Chaque formation a un code alphanumérique, possède un libellé, un nombre maximum de places et un prix de base soumis dans certains contextes à des réductions (cf. ultra). Une formation peut être sponsorisée ou non par la Région Wallonne.

Parmi les formations choisies par un inscrit, il devra sélectionner une formation principale.

**Cet exercice n'est plus dirigé : toutes les démarches ne sont plus nécessairement décrites. À vous de réfléchir à la découpe de votre code ; parfois, il vous faut écrire ou compléter des méthodes et propriétés autres que celles explicitées.**

Voici le diagramme de classes correspondant :



Certains participants bénéficient d'un tarif réduit sous certaines conditions relatives à leur statut, leur domicile et/ou à la localisation de l'institution dans laquelle ils travaillent :

- Les chômeurs bénéficient d'une réduction de 100%.
- A priori, les travailleurs ne bénéficient d'aucune réduction mais ceux qui travaillent dans une institution du grand Namur ont droit à 20% de réduction.
- Les enseignants constituent un cas particulier : ils bénéficient normalement d'une réduction de 50% mais celle-ci passe à 70% si l'établissement dans lequel ils travaillent est situé dans le grand Namur.
- En plus de cela, le pourcentage de réduction est augmenté de 25% (sans dépasser les 100% toutefois) lorsqu'il s'agit d'un participant domicilié en Région Wallonne et que la formation est sponsorisée par la Région Wallonne.

## DESCRIPTION DES CLASSES

- Une classe Utilitaires reprend :
  - Une méthode `GrandNamur` qui indique si un code postal correspond au « grand Namur » ;
  - Une seconde méthode `RegionWallonne` qui indique si un code postal correspond à la Région Wallonne.

*Note. Le « grand Namur » correspond aux localités dont le code postal commence par « 50 ». La Région Wallonne recouvre les codes postaux entre 1300 et 1499 et entre 4000 et 7999.*
- La classe Institution permet de modéliser des entreprises et autres institutions.
- La classe Formation permet de modéliser des formations.
  - Prévoyez une propriété Prix qui permet de modifier la valeur. Aucun prix ne peut être inférieur à 100 euros. Si la nouvelle valeur est inférieure à 100, le prix est initialisé à 100.
  - Dans le constructeur, la valeur par défaut pour le sponsoring par la Région Wallonne est `true` ; il n’y a pas encore d’inscrits à la création de la formation. Assurez-vous de passer par la propriété `Prix`.
  - Une constante `NB_PLACES_MAX` est déclarée ; sa valeur vaut 25.
  - Une propriété Informations donne accès à une chaîne de caractères composée de la description de la formation (code, libellé et mention « (RW) » pour les formations sponsorisées par la Région Wallonne).

Exemples : `JSC007 - Introduction à Javascript`  
`NET042 - Trouver l’info sur internet (RW)`
  - La méthode `ToString` renvoie la description de la formation ainsi que le nombre actuel d’inscrits à celle-ci entre crochets.

Exemples : `JSC007 - Introduction à Javascript [7 inscrit(s)]`  
`NET042 - Trouver l’info sur internet (RW) [1 inscrit(s)]`

*Note : aucun calcul ne sera admis dans `ToString`.*
- La classe abstraite Inscrit permet de modéliser les participants.
  - Comme chaque inscrit est obligé de choisir une formation principale, la création d’un objet doit se faire avec une formation qui, par défaut, est la principale. Par la suite, l’inscrit peut modifier ce choix s’il s’inscrit à d’autres formations. Réfléchissez bien à la meilleure manière de modéliser cette demande (plusieurs solutions sont possibles).
  - `NB_MAX_FORMATIONS_SUIVIES` représente le nombre maximal de formations auxquelles un participant peut s’inscrire (actuellement 5).

- Le constructeur général permet de créer un nouveau participant avec la première formation qu'il a choisie et qui est par défaut sa formation principale. Si la référence de la formation est `null`, levez une exception via l'instruction suivante (placée au bon endroit) :  
  
`throw new Exception("Il faut une formation principale !");`
- La méthode `ToString` renvoie le nom de l'inscrit suivi du code postal de son domicile entre parenthèses.
- Ajoutez une propriété `FormationPrincipale` en lecture et en écriture qui utilise un code de formation ; en lecture : elle renvoie le code de la formation principale ; en écriture : si le code donné correspond à une des formations choisies par l'inscrit, celle-ci devient sa formation principale.
- Prévoyez une méthode `AjoutFormation` qui ajoute une formation. Elle accepte un paramètre booléen facultatif si la formation ajoutée est la (nouvelle) principale. N'oubliez pas de tester si l'inscrit n'a pas déjà atteint son nombre maximum de formations, si la formation peut encore accepter un inscrit et si l'inscrit ne l'a déjà pas choisie. S'il l'a déjà choisie comme non principale et que la formation devient principale, modifiez son statut.
- Prévoyez une méthode `AjoutFormations` qui ajoute plusieurs formations au participant. Si le participant a déjà choisi son nombre maximal de formations, on ne fait rien ; s'il en choisit de trop, on ne prend que les premières à ajouter. Si une formation n'offre plus de place, on ne l'ajoute pas. (Limite : nous n'envisageons pas l'aspect principal ou non pour les formations demandées.)
- Écrivez une propriété abstraite `PourcRéductionBase`, qui indique le pourcentage de réduction accordé au participant sur le prix des formations non sponsorisées par la Région Wallonne.
- Écrivez une méthode `Coût` qui renvoie le coût d'une formation (en tenant compte des réductions spécifiques aux formations sponsorisées par la Région Wallonne).
- Écrivez une méthode `CoûtTotal` qui renvoie le coût de l'ensemble des formations choisies.
- Écrivez la méthode `FicheInformations` qui renvoie la facture de l'inscrit au format décrit ci-dessous.

Exemple : *Inscrit : Pedro Dupond (5005)*  
*JSC007 - Introduction à Javascript*  
*UML042 - UML pour les nuls (RW)*  
*Total à payer : 54.50 Euros*

- La classe **Chômeur** permet de modéliser un participant inscrit au chômage.

- La classe **Travailleur** permet de modéliser un participant non-chômeur.
  - Prévoyez le constructeur général en faisant attention au fait que le travailleur n'est pas toujours lié à une institution.
  - Créez une propriété **Occupation** qui renvoie le métier et le nom de l'institution si l'institution est mémorisée.  
Exemples : *employé chez BiApi*  
*indépendant*
  - La méthode **ToString** renvoie le nom de l'inscrit suivi du code postal de son domicile entre parenthèses ; si le travailleur est affilié à une institution, le nom et le code postal de celle-ci complète ces informations.  
Exemples : *John Lindépendant (4300) - indépendant*  
*Hubert Lemployé (5000) - employé chez BiApi*
- La classe **Enseignant** permet de modéliser un enseignant ou une enseignante.  
Dans le constructeur, si l'institution a une référence **null**, on lève une exception.  
La propriété **Occupation** renvoie la chaîne composée du métier, du nom de l'institution et du domaine entre crochets.  
Exemple : *enseignant chez Hénallux - [Domaine : orienté objet]*

**Le client souhaite connaître combien lui rapporte une formation suivant les inscrits qui l'ont choisie. Que faire ?**