

# Initialisation



- Lancer la VM
  - login : **siad** - password : **javaeesiad**
- Ouvrez une console et rendez vous dans le repertoire de dev
  - Créez un repertoire workspaces et placez vous dedans
    - Astuce : `mkdir workspaces && cd $_`
- Il s'agit maintenant de cloner le repository de source (hébergé par github)
  - Exécutez `git clone https://github.com/ptitbob/siad_m2_ws.git`
  - Placez vous dans le repertoire `siad_m2_ws`
  - Exécutez `mvn clean compile` (il devrait télécharger le web)

Pour faire les mise à jour pour chaque cours (si la VM n'a pas été réinitialisée) :

- `git pull origin`
- `mvn clean compile`

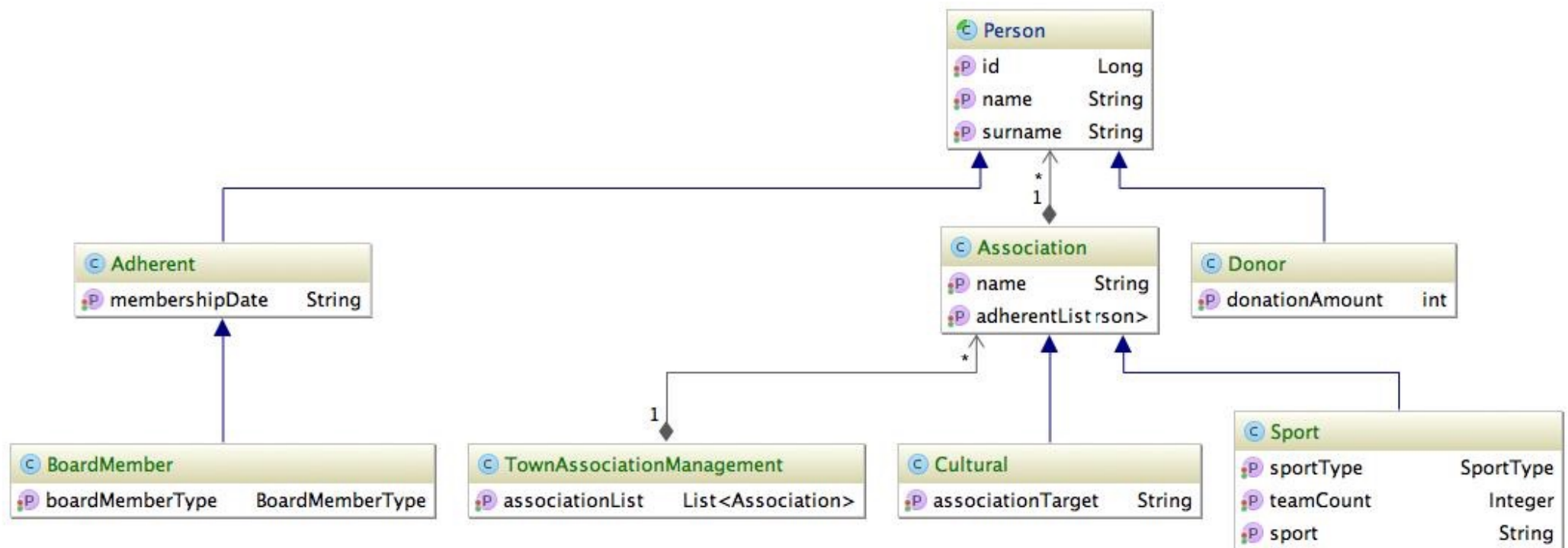
# Les outils du cours

- VM Linux (xubuntu)
- Java 8
- IDEs
  - IntelliJ
  - Netbeans
- Maven
- Serveur d'application
  - Glassfish
  - Wildfly
  - Jetty (container de servlet)
- Git
- shell



**maven**

# JAXB - TD 1



Powered by yFiles

Rappelez vous :

- un bean a toujours un constructeur vide
- Le code est en anglais
- les commentaires et javadoc en français

# JAXB - TD 1



A partir de la classe de test `TownAssociationTest` et de la méthode principale...

- Suivez les demandes exprimées en commentaire
  - Dans la classe de test et dans chaque bean
- Sur la classe `Person`
  - Ajouter un `id` qui sera obligatoirement en attribut
- Sur la classe `BoardMember`
  - Le type de poste devra être en attribut
- La classe association devra présenter un champs nombre d'adhérents non sérialisé
- Créer une classe `Address`, chaque personne devra la présenter
- Créer un schéma pour chaque package avec comme namespace
  - base (`fr.univ.blois.siad.m2.ws.jaxb.td01`) : **town**
  - `Person` : **adh**
  - `Club` : **clb**
- Pour l'énum `SportType`, il devra être serialisé sous forme d'entier
- Générer le fichier
- *Si il vous reste du temps, amusez vous à générer les schémas*