Inngéniérie du WEB sémantique
POLYTECH. 5ième année
UNIVERSITÉ D'AIX-MARSEILLE
Année universitaire 2017/2018

Pierre Drap
Odile Papini

Feuille de T. P. 1 : Manipuler une ontologie en java

Préliminaires

Se connecter sous Linux

Télécharger lles fichiers compréssés dans AMUBox,
dossier **POUR-TP-ING-WEBSEM** à partir de l'URL :
https ://amubox.univ-amu.fr/index.php/s/NydVUndV1deL8Uh

Celle-ci contient :
- un fichier pizza_1.owl
- un dossier **library-OWL-API** : qui contient de les librairies utiles pour le TP

Créer un nouveau projet java sous Eclipse ou NetBeans

Sauvegarder :
- toutes les librairies du dossier library-OWL-API
- la documentation OWL-API : $http://owlapi.sourceforge.net/documentation.html$ qui contient un tutoriel sor $OWL-API$ et des exemples de code java.
- le tutoriel : $http://owlapi.sourceforge.net/owled2011\_tutorial.pdf$

ATTENTION : lorsqu'on travaille avec un fichier owl crée avec Protégé sauvegardé dans un répertoire de l'utilisateur, vérifier que l'IRI (sous Linux) du fichier est bien : $file:///Users/\ chemin\ jusqu'au\ fichier\ /nom\_fichier.owl$

Création d'une classe java $OWLAPI\_StepByStep$, la classe $OWLAPI\_StepByStep$ contient une classe java $PIZZA$ qui contient les constantes : nom de l'ontologie, IRI, espace de noms.

La philosophie de $OWL-API$: une ontologie (ou une OWL ontologie) est une interface de type $OWLOntology$ qui modélise des axiomes ($OWLAxioms$). Une ontologie a un IRI et des méthodes qui lui sont associées. Dans la suite *ontology* désigne une variable de type $OWLOntology$. Lire attentivement les tutoriels avant de commencer à écrire les méthodes suivantes.

— Ecrire une méthode $read(OWLAPI\_StepByStep.ontologyFileName)$ qui retourne une ontologie de type $ontology$ à partir du fichier $pizza\_1.owl$.

— Ecrire une méthode une méthode $printClasses(ontology)$ qui affiche les classes d'une ontologie passée en paramètre.

— Ecrire une méthode une méthode $printObjectProperty(ontology)$ qui affiche les propriétés d'une ontologie passée en paramètre.

— Ecrire une méthode une méthode $printDataTypeProperty(ontology)$ qui affiche les propriétés de type de données d'une ontologie passée en paramètre.

— Ecrire une méthode une méthode $printAtoms(ontology)$ qui affiche les atomes d'une ontologie passée en paramètre.

— Ecrire une méthode $save(OntoData.ontology, OntoData.manager, savedOntologyFileName\_A)$ qui sauvegarde l'ontologie dans le fichier $pizza\_1\_A.owl$

— Ecrire une méthode $saveInOWLXML(OntoData.ontology, OntoData.manager, savedOntologyFileName\_B)$ qui sauvegarde l'ontologie au format XML dans le fichier $pizza\_1\_B.owl$

— Ecrire une méthode $addingClassesAndComment(OntoData.ontology, OntoData.manager)$ qui ajoute une classe et un commentaire et sauvegarde la nouvelle ontology dans dans le fichier $pizza\_1\_C.owl$. Ajouter la classe $PizzaDuSamediSoir$ qui est une sous-classe de $PizzaDuSoir$ et $PizzaDuSoir$ est une sous-classe de la classe $LSISPizza$ dans l'ontologie contenue dans le fichier $pizza\_1.owl$

— Ecrire une méthode $addingIndividuals(OntoData.ontology, OntoData.manager)$ qui ajoute un individu. Ajouter une instance $MaRoyaleSansOlive$ qui est une instance de $PizzaDuSamediSoir$ classe de l'ontologie contenue dans le fichier $pizza\_1\_C.owl$.

— Ecrire une méthode $addingDataPropertyToIndividuals(OntoData.ontology, OntoData.manager)$ qui ajoute une propriété de donné (data property). Ajouter un diamètre de 33.3 cm à $MaRoyaleSansOlive$

— Ecrire une méthode $walkThroughOntology(OntoData.ontology, OntoData.manager)$ qui parcours (visite) l'ontologie. (Voir la notion de visiteur d'ontologie dans les tutoriels)

— Ecrire une fonction $main$ pour tester toutes ces méthodes.