

## TD5 Application Concours UI/Métier/Dao avec Spring (projet spring-concours)

Il s'agit de développer une application de gestion des candidats selon un découpage en couches avec Spring. La couche Dao est fournie sous la forme d'un projet Maven (**dao-concours**) qui contient deux implémentations, l'une avec accès à un fichier JSON (TD2), l'autre exploitant une base de données MySQL (TD3).

- Installer la base de données (fichier db\_concours.sql) Wamp/phpMyAdmin
- Fichiers JSON à copier dans le dossier du projet Java/Maven (emplacement pom.xml)
- Ouvrir le projet Maven de la couche DAO dans Eclipse
- Installer le projet de la couche DAO dans le dépôt Maven
- Créer le projet Maven (spring-concours) de la couche UI/Metier, voir ci-dessous

**Créer un projet Maven (spring-concours) :** (gID) pta.sagi / (aID) spring-concours / 0.1

**Ajouter les dépendances:**

pta.sagi / dao-concours [couche DAO]  
junit/junit/4.11  
org.springframework/spring-context/4.3.7.RELEASE

**Travail à faire :** les couches seront liées par le framework Spring (classe de config. Spring).

développer la couche Métier + tests métier (voir le diagramme de classes)

développer une couche UI console (cf exemple de sortie console ci-dessous)

développer une interface GUI JavaFX (cf fin TD4)

Pour l'application avec UI console, écrire une classe exécutable qui crée le contexte Spring et lance l'exécution.

```
public class ExecConcours {  
    public static void main(String[] args)  
    {  
        AnnotationConfigApplicationContext ctx;  
        ctx=new AnnotationConfigApplicationContext(ConfigSpring.class);  
        ctx.getBean(IUserInterfaceConcours.class).run();  
    }  
}
```

**Couche Métier :**

- Fournit les candidats d'une ou plusieurs régions (paramètres variables p4/22 Eléments de Syntaxe Java)
- Trie les résultats par score décroissant
- Fournit les N meilleurs par région (getTop)
- On peut définir N (topSize)
- Permet l'ajout d'un candidat

**Couche UI Console :** interactions console avec l'utilisateur (cf exemple TD4 p4/11 et Synthèse Java p17/22). Un exemple d'exécution ci-contre

Afficher un prompt :

0:Q 1:Liste/region 2:Top3/region 3:Add

```
0:Q 1:Liste/region 2:Top3/region 3:Add  
2  
Région=?  
2  
Les candidats:  
nom=Beatrice Ewing, num=756699, score=90.9, region=2  
nom=Eaton Nieves, num=776099, score=87.0, region=2  
nom=Dakota Rhodes, num=505199, score=84.6, region=2  
0:Q 1:Liste/region 2:Top3/region 3:Add  
3  
Saisir format [Nom;num;score;region]  
[John Moodle;641235;88.5;2]  
0:Q 1:Liste/region 2:Top3/region 3:Add  
2  
Région=?  
2  
Les candidats:  
nom=Beatrice Ewing, num=756699, score=90.9, region=2  
nom=John Moodle, num=641235, score=88.5, region=2  
nom=Eaton Nieves, num=776099, score=87.0, region=2  
0:Q 1:Liste/region 2:Top3/region 3:Add  
0  
Bye!
```

**Tests :** on peut utiliser Spring pour créer les couches pour les tests également.

```
// Tests (on crée un contexte Spring comme variable de classe)

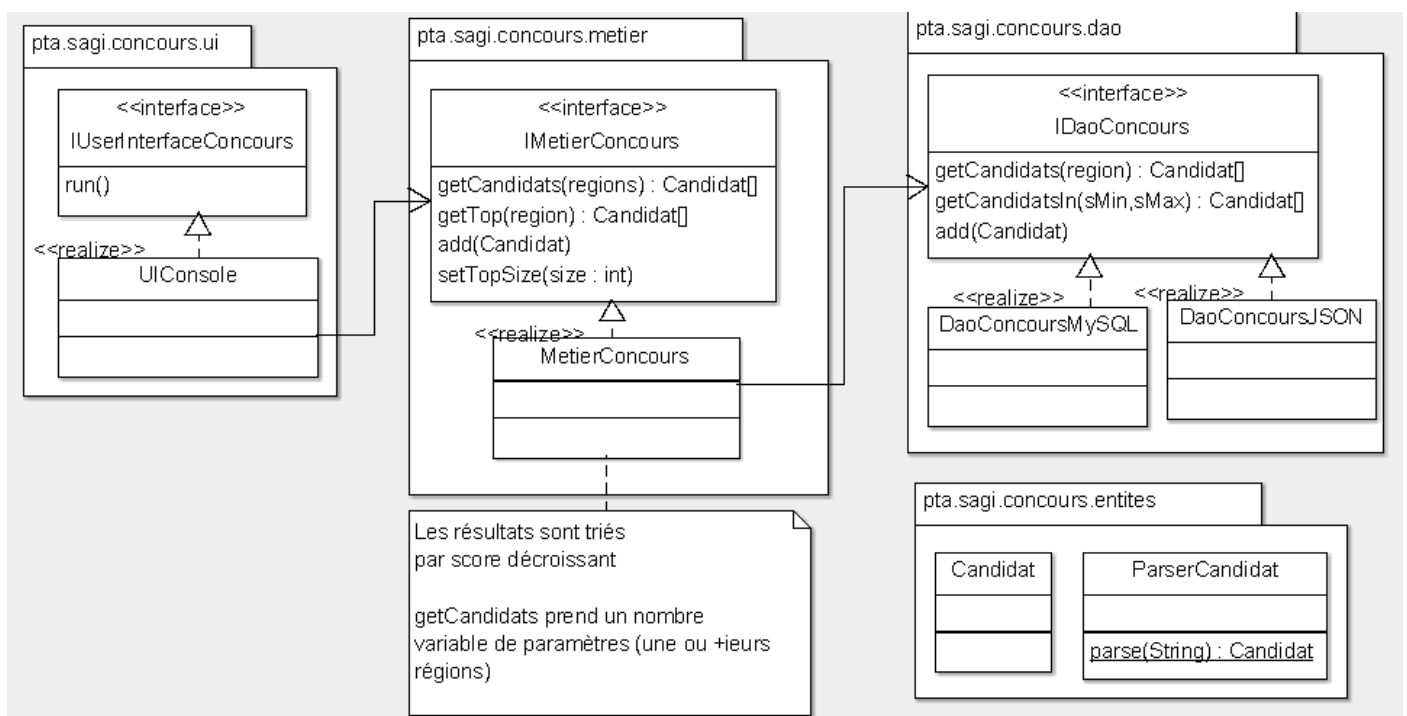
public class TestConcours
{
    private static AnnotationConfigApplicationContext ctx;

    @BeforeClass
    public static void setUp(){
        ctx=null;
        try{ ctx=new AnnotationConfigApplicationContext(ConfigSpring.class);}
        catch(Exception ex){ fail("Problème Spring");}
    }

    @AfterClass
    public static void tearDown(){ if(ctx!=null) ctx.close(); }

    @Test
    public void testDao()
    {
        IDaoConcours dao;
        dao=ctx.getBean(IDaoConcours.class); // obtient le bean Dao
        ...
    }
}
```

## Diagramme UML de l'application Concours



## Interface GUI de l'application Concours avec JavaFX

En reprenant les techniques vues aux TD3 et TD4, développer l'interface utilisateur sous forme de fenêtre JavaFX permettant

- De lire les candidats par région
- De lire le Top d'une région

Selon le temps : contrôles pour permettre de définir la taille du Top et l'ajout d'un candidat (Parser)

