Rapport UML : sonnarlint

Nassim Benchikh

February 2024

1 SonnarLint

SonnarLint est une extension qui permet de détecter des bugs relatif aux vulnérabilités du code et propose des soltion pour les ajuster.

Les erreurs de code sont détectées au même moment ou l'on code à l'instar des verificateurs d'écriture Word (spell-checker)

SonnarLint pointe directement vers où l'erreur se situe, et les trie en 3 couleurs:

- Suppression, renommage, compléter, ..
- Ajout, définition de consatantes, problèmes de complexité ...
- Changements (ex: change visibility), proposition d'améliorations, ...

2 Changements les plus persistants en utilisant SonnarLint

Change the visibility of this constructor to "protected"

Les constructeurs des classes abstraits sont uniquement accessibles dans leurs sous-classes, les mettre à **public** ne sert donc a rien. Il suffit de les mettre à **protected**.

Remove useless brackets

Concerne les lambdas expréssions pour la plus art, nécissite pas de changements.

Utilisation de variables statique

Cela a pour but de representer des messages d'erreurs et des propriétés qui ont chnagées lorsque le code est redondant

3 Changements et questionnements

Import inutils

Il est plus judiceux de les commenter

Unused variable

Suppression des variables inutles, cela a été fait et l'application fonctionne toujours

AEntity

- Transformation de la lambda expression en boucle for-each
- Itération sur les entrySet() car plus efficace pour obtenir les valeurs que sur les keySet()

```
public void addAllStyle(Map<String, String> keyValue)
   {
    Contract.check(keyValue != null);
    for (Map.Entry<String, String> entry : keyValue.
       entrySet()) {
        getDiagram().getStyle().addStyle(getStyleId(),
            getId(), entry.getKey(), entry.getValue())
           ;
    }
}
```

ARelationGraphic

Utilisaion d'une condtane **C_NULL** pour le message d'erreur : "c must not be null !"

AUseCaseVisitor

5 méthodes Méthodes **visit** sans corps.

AbstractPackageVisitor

Import de **clazz.view.ClassDiagram** detécté comme inutile par SonnarLint. L'import a été gardé comme même pour la suite.

Activity

Méthodes non complétées
Méthode accept : visitor.viir(this) a été commenté par l'ancien groupe ?

```
public void addAllStyle(Map<String, String> keyValue)
   {
                // TODO Auto-generated method stub

        }

        @Override
        public void removeStyle(String key) {
                // TODO Auto-generated method stub

        }

        @Override
        public void clearStyle() {
                // TODO Auto-generated method stub

        }

        public void setParent(IActivity parent) {
                IActivity old = this.parent;
                IActivity news = parent;
                this.parent = parent;
                parent.getActivity().add(this);
                firePropertyChange(
                   PARENT_CHANGED_PROPERTY_NAME, old,
                   news);
        }

        @Override
        public void accept(ISequenceVisitor visitor) {
                // TODO a faire
                // visitor.visit(this);
        }
```

Definir une Exception au lieu d'utiliser une Exception generique (pas obligatoire pour le moment) :

..... throws Exception {
else { throw new Exception ("...")
}

ActivityGraphic

Suppression des champs private IObjectGraphic src et private IObjectGraphic dst car non utilisés.
Code mal structuré avec quelques problèmes de complexités (pas de modification faites)

ActivityGraphicController

MenuItem addRelationMI = getAddRelationMI() non ustilisé (champ supprimé)

Remplecement de valeuOf par parseDouble qui rend le code plus éfficace.

Proposition par SonnarLint d'utilisation de **Optional** sur `diagramController.getObjects().values().stream().findFirst()`.

Partie de code comentée par l'ancien groupe :

```
/* addRelationMI.setOnAction(new EventHandler<
   ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                    diagramController.createRelation(
                       model);
            }
        });*/
```

ActorGraphic

Rajout de **static** aux constantes de classe

ActorGraphicController

Appel de a méthodes commentés par l'ancien groupe

```
//manage name modification
      act.addPropertyChangeListener(
         NAME_CHANGED_PROPERTY_NAME, evt -> {
          //ag.draw();
          ag.setName();
      });

      //manage style modification
      act.addPropertyChangeListener(
         STYLE_CHANGED_PROPERTY_NAME, evt -> {
          setStyle((IStyle) evt.getNewValue(), false
             );
          //ag.draw();
          ag.setStyleToActor();
      });
```

ActorTest

2 méthodes du code commentéés par l'ancien groupe

```
public class ActorTest {
prj = new Project("a");
.
.
.
  @Before
  public void setUp() throws Exception {
      .
      .
      .

  }

  @Test
  public void testGetAddRemoveGeneralization()
     throws Exception {

      /*

      .
      .
      .
```

```
        */
    }

    @Test
    public void testGetAddRemoveAssociation() throws
       Exception {
      /*

         .
         .
         .

         */
    }


}
```

Aggregation

Import détectés cmme inutile par SonnarLint. (suppression des imports)

ArrowBody

Oprimisation du code en mettant une constante d'erreur : **P_NULL = "p must not be null"**

Attribute

Manque de **default** a **switch** (laissé comme tel)

ClassController

ViewRelation viewRelation = new ViewRelation(r, diagram); pas utilisé (mis en commentaires)

SonnarLint : Replace this lambda with method reference 'ClassController.this::removeObjectEntity'. (laisser come tel pour instant.

4 Remarques

- Dans la classe AEntityRelation : SonarLint demnede le changemet de **relations** en private or il faut le laisser comme tel (protected)
- Tous signalement de type Remove useless brackets/parentheses ou autre n'ont pas étés modifiés.
- Méthodes avec de TO-DO non touchées por le moment.
- Manque de default dan sla plus part des switch
- les lambdas ecpression ont étées laissées comme tel, sauf dans une méthide ou ele a été remplacée par une boucle for-each (sa fonctionne toujours)
- call Optional non touché pour l'insatant