

# Rapport<sub>T</sub>*P<sub>r</sub>ca*

chahi26ben

October 2020

## 1 Partie 1 : Extraire les artefacts

### 1.1 Exercice 1 : Extraire les OBEs

Dans cette exercice vous réaliserez l'extraction des artefacts de la LPL. A partir du code sources des produits (à récupérer sur le moodle du cours), vous devez identifier les différents artefacts en vous appuyant le méta-modèle figure 2. Ce méta-modèle définit ce que sont les artefacts.

- 1. Pour les produits P1, à la main, identifiez les Object-oriented Building Elements (OBEs) de la classe Line.java en suivant le méta-modèle figure 2 et réalisez les diagrammes d'instance des OBEs de ce fichier.

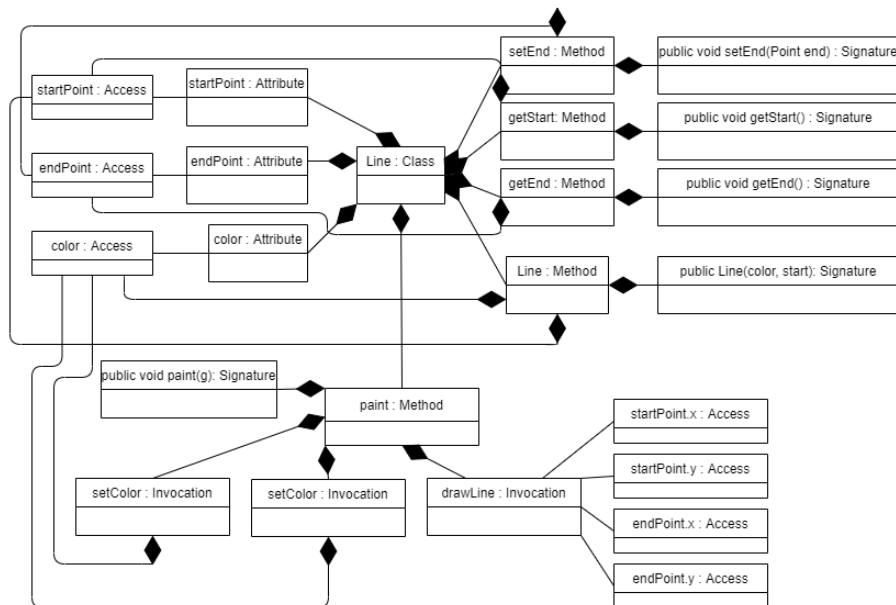


Figure 1: Object-oriented Building Elements

- 2 & 3. Réalisez maintenant un programme Java qui permet de réaliser l'identification et l'instanciation de ce méta-modèle. Pour cela, utilisez SPOON. Commencez par modéliser les classes du méta-modèle dans votre

programme. Puis, avec SPOON, parcourez l'AST afin d'instancier vos classes OBE.

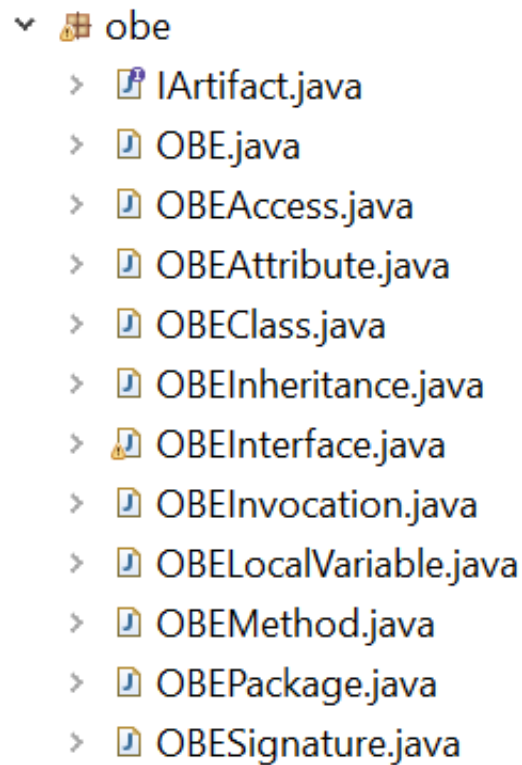


Figure 2: les classes du méta-modèle

## 1.2 Exercice 2 : Définir les artefacts

Maintenant que le Méta-Modèle de la LPL peut être instancié dans votre programme Java, vous allez récupérer les artefacts à proprement parler.

- 1 & 2. Rajoutez une interface Java à votre programme, nommez cette interface IArtefact et implémentez là dans les classes Class, Attribute et Method.

```
1 package obe;
2
3 public interface IArtefact {
4
5     String getName();
6     int getIdentifiant();
7 }
8
```

Figure 3: interface IArtefact

- 3. Complétez le corps des méthodes de IArtefact dans les classes Class, Attribute et Method.

```
@Override
public String getName() {
    return this.name;
}

@Override
public int getIdentifiant() {
    return getName().hashCode();
}
```

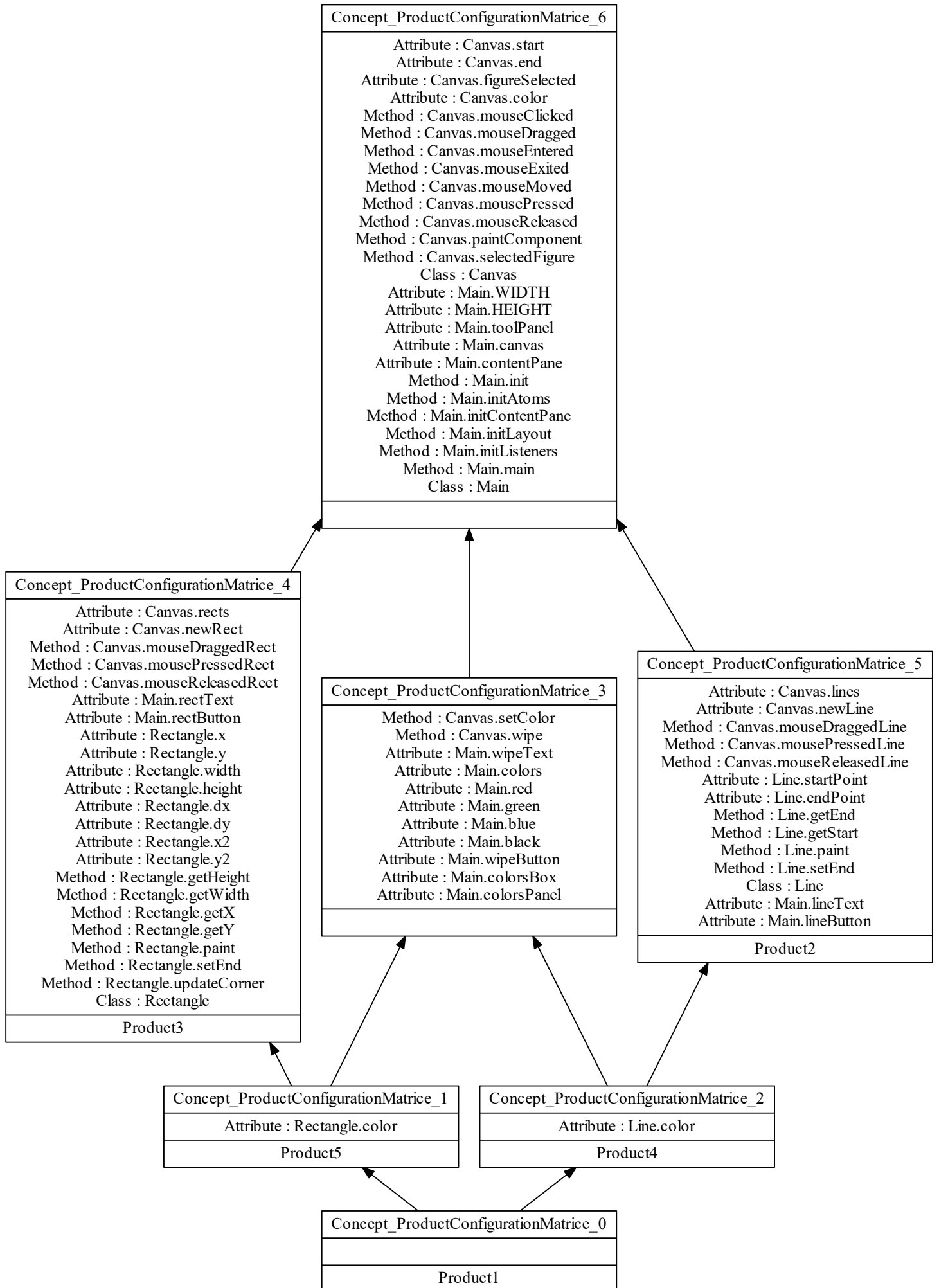
Figure 4: le corps des méthodes de IArtefact

- 4 & 5. Ajoutez à votre programme une fonction qui renvoie la liste des artefacts qui composent un produit. Astuce : le type IArtefact vous permet de facilement identifier les artefacts parmi l'ensemble des OBEs.

```
public static List<IArtefact> getIArtefact(){
    List<IArtefact> listArtefact = new ArrayList<IArtefact>();
    List<CtPackage> packag = model.getElements(new TypeFilter<CtPackage>(CtPackage.class));
    for(CtPackage newPack : packag) { // récupéré tous les packages :
        OBEPackage newPackage = new OBEPackage("Package : "+newPack.getSimpleName());
        for(CtType<?> type : model.getAllTypes()) { // récupéré toutes les classes :
            if (type instanceof CtClass) {
                OBEClass newClasse = new OBEClass("Class : "+type.getSimpleName());
                List<CtField<?>> listField = type.getFields();
                for (CtField<?> field : listField) { // récupéré tous les attributs :
                    OBEAttribute newAttribute = new OBEAttribute("Attribute : "+type.getSimpleName()+"."+field.getSimpleName());
                    newClasse.getBelongToClass().add(newAttribute);
                    listArtefact.add(newAttribute);
                }
            }
            for(CtMethod<?> newMethod : type.getMethods()) { // récupéré toutes les méthodes :
                OBEMethod newM = new OBEMethod("Method : "+type.getSimpleName()+"."+newMethod.getSimpleName());
                List<CtInvocation> listInvocation = newMethod.getElements(new TypeFilter<CtInvocation>(CtInvocation.class));
                for(CtInvocation invok : listInvocation) { //Récupéré toutes les invocations :
                    OBEInvocation newInvocation = new OBEInvocation("Invocation : "+getSimpleName(invok));
                    newM.getInvocations().add(newInvocation);
                }
            }
            List<CtLocalVariable> listLV = newMethod.getElements(new TypeFilter<CtLocalVariable>(CtLocalVariable.class));
            for(CtLocalVariable lv : listLV) { //Récupéré toutes les variables locales :
                OBELocalVariable newLV = new OBELocalVariable("Local variable : "+lv.getSimpleName());
                newM.getLocalVariables().add(newLV);
            }
            newClasse.getMethod().add(newM);
            listArtefact.add(newM);
        }
        newPackage.getListClass().add(newClasse);
        listArtefact.add(newClasse);
    }
    if (type instanceof CtInterface) {
        OBEInterface newClasse = new OBEInterface("Interface : "+type.getSimpleName());
        newPackage.getListInterface().add(newClasse);
    }
}
```

Figure 5: fonction qui renvoie la liste des artefacts

- 6. Téléchargez rca-explorer (disponible sur moodle). Suivez les instructions de la vidéo pour créer un concept formel à partir du PCM.



- 7. Exécutez AFC de rca-explorer sur le PCM pour obtenir le treillis de concept. A partir du Treillis, donnez la liste des artefacts communs et optionnels de la LPL.

1. artefacts communs

```
les artefacts communs :

Attribute : Canvas.start
Attribute : Canvas.end
Attribute : Canvas.figureSelected
Attribute : Canvas.color
Method : Canvas.mouseClicked
Method : Canvas.mouseDragged
Method : Canvas.mouseEntered
Method : Canvas.mouseExited
Method : Canvas.mouseMoved
Method : Canvas.mousePressed
Method : Canvas.mouseReleased
Method : Canvas.paintComponent
Method : Canvas.selectedFigure
Class : Canvas
Attribute : Main.WIDTH
Attribute : Main.HEIGHT
Attribute : Main.toolPanel
Attribute : Main.canvas
Attribute : Main.contentPane
Method : Main.init
Method : Main.initAtoms
Method : Main.initContentPane
Method : Main.initLayout
Method : Main.initListeners
Method : Main.main
Class : Main
```

Figure 6: artefacts communs

2. artefacts optionnels

```
les artefacts optionnels :  
  
Attribute : Line.color  
Attribute : Rectangle.color  
Attribute : Canvas.rects  
Attribute : Canvas.newRect  
Method : Canvas.mouseDraggedRect  
Method : Canvas.mousePressedRect  
Method : Canvas.mouseReleasedRect  
Attribute : Main.rectText  
Attribute : Main.rectButton  
Attribute : Rectangle.x  
Attribute : Rectangle.y  
Attribute : Rectangle.width  
Attribute : Rectangle.height  
Attribute : Rectangle.dx  
Attribute : Rectangle.dy  
Attribute : Rectangle.x2  
Attribute : Rectangle.y2  
Method : Rectangle.getHeight  
Method : Rectangle.getWidth  
Method : Rectangle.getX  
Method : Rectangle.getY  
Method : Rectangle.paint  
Method : Rectangle.setEnd  
Method : Rectangle.updateCorner  
Class : Rectangle  
Method : Canvas.setColor  
Method : Canvas.wipe
```

Figure 7: artefacts optionnels1

```
Attribute : Main.wipeText
Attribute : Main.colors
Attribute : Main.red
Attribute : Main.green
Attribute : Main.blue
Attribute : Main.black
Attribute : Main.wipeButton
Attribute : Main.colorsBox
Attribute : Main.colorsPanel
Attribute : Canvas.lines
Attribute : Canvas.newLine
Method : Canvas.mouseDraggedLine
Method : Canvas.mousePressedLine
Method : Canvas.mouseReleasedLine
Attribute : Line.startPoint
Attribute : Line.endPoint
Method : Line.getEnd
Method : Line.getStart
Method : Line.paint
Method : Line.setEnd
Class : Line
Attribute : Main.lineText
Attribute : Main.lineButton
```

Figure 8: artefacts optionnels2

### 1.3 Exercice 3 : Extraire la variabilité et les caractéristiques

- 2. Toujours à partir du treillis, identifiez les artefacts co-occurents. Pour chaque groupes d'artefacts co-occurents, listez l'ensemble de ses artefacts en donnant le type et son nom. (ex. Class:Line, Attribut:Color, etc).

Concept_ProductConfigurationMatrice_6
Attribute : Canvas.start Attribute : Canvas.end Attribute : Canvas.figureSelected Attribute : Canvas.color Method : Canvas.mouseClicked Method : Canvas.mouseDragged Method : Canvas.mouseEntered Method : Canvas.mouseExited Method : Canvas.mouseMoved Method : Canvas.mousePressed Method : Canvas.mouseReleased Method : Canvas.paintComponent Method : Canvas.selectedFigure Class : Canvas Attribute : Main.WIDTH Attribute : Main.HEIGHT Attribute : Main.toolPanel Attribute : Main.canvas Attribute : Main.contentPane Method : Main.init Method : Main.initAtoms Method : Main.initContentPane Method : Main.initLayout Method : Main.initListeners Method : Main.main Class : Main

Figure 9: artefacts co-occurents1



Concept_ProductConfigurationMatrice_4
Attribute : Canvas.rects Attribute : Canvas.newRect Method : Canvas.mouseDraggedRect Method : Canvas.mousePressedRect Method : Canvas.mouseReleasedRect Attribute : Main.rectText Attribute : Main.rectButton Attribute : Rectangle.x Attribute : Rectangle.y Attribute : Rectangle.width Attribute : Rectangle.height Attribute : Rectangle.dx Attribute : Rectangle.dy Attribute : Rectangle.x2 Attribute : Rectangle.y2 Method : Rectangle.getHeight Method : Rectangle.getWidth Method : Rectangle.getX Method : Rectangle.getY Method : Rectangle.paint Method : Rectangle.setEnd Method : Rectangle.updateCorner Class : Rectangle

Figure 10: artefacts co-occurents2

Concept_ProductConfigurationMatrice_3
Method : Canvas.setColor Method : Canvas.wipe Attribute : Main.wipeText Attribute : Main.colors Attribute : Main.red Attribute : Main.green Attribute : Main.blue Attribute : Main.black Attribute : Main.wipeButton Attribute : Main.colorsBox Attribute : Main.colorsPanel

Figure 11: artefacts co-occurents3

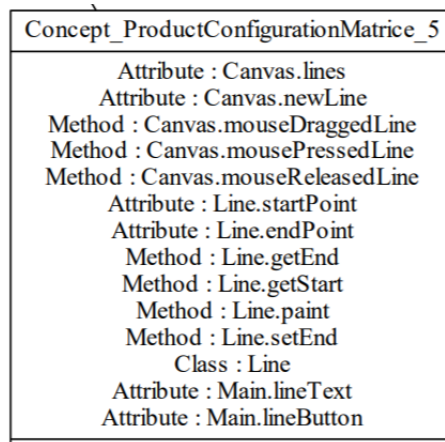


Figure 12: artefacts co-occurents4

- 3. Explorez le code source associé à vos groupes: si nous devons associer chaque groupe1 d'artefacts co-occurents à une caractéristique, est-ce que cela serait pertinent?

Non, ceci n'est pas pertinent car il y'aura des concept co-occurent qui ne seront pas reliable à une des caractéristique.

Donnez un groupe pour lequel l'association 1-1 avec une caractéristique est pertinente. Et donnez un groupe pour lequel cette association ne vous semble pas normale. Argumentez votre décision.

il n'ya pas de groupes pour lequel l'association 1-1 avec une caractéristique est pertinente. Pour le concept : " Concept\_ProductConfigurationMatrice\_6 ", on peut appercevoir qu'aucune des caractéristique ne peut correspondre.

## 2 Partie 2 : L'identification des caractéristiques

### 2.1 Exercice 4 : Couplage Structurelle

Comme nous l'avons vu en cours, les groupes d'artefacts co-occurents peuvent cacher l'implementation de plusieurs caractéristiques. A partir des résultats de l'exercice précédant, vous allez implémenter l'approche vu en cours pour identifier les groupes minimaux d'artefacts.

- Analysez les OBEs de votre LPL pour obtenir les relations de dépendances structurelle entre les artefacts appartenant à un même groupe d'artefacts co-occurents. Produisez le graphe des dépendances pour ce groupe d'artefacts. Utilisez pour cela les classes et leurs relations que vous avez instancié dans l'exercice 1.

```

digraph G {
  "Attribute : Canvas.figureSelected" -> "Method : Canvas.mouseDragged"
  "Attribute : Canvas.figureSelected" -> "Method : Canvas.mousePressed"
  "Attribute : Canvas.figureSelected" -> "Method : Canvas.mouseReleased"
  "Attribute : Canvas.figureSelected" -> "Method : Canvas.selectedFigure"
  "Method : Canvas.mouseClicked" -> "Class : Canvas"
  "Method : Canvas.mouseDragged" -> "Class : Canvas"
  "Method : Canvas.mouseEntered" -> "Class : Canvas"
  "Method : Canvas.mouseExited" -> "Class : Canvas"
  "Method : Canvas.mouseMoved" -> "Class : Canvas"
  "Method : Canvas.mousePressed" -> "Class : Canvas"
  "Method : Canvas.mouseReleased" -> "Class : Canvas"
  "Method : Canvas.paintComponent" -> "Class : Canvas"
  "Method : Canvas.selectedFigure" -> "Class : Canvas"
  "Attribute : Canvas.start" -> "Class : Canvas"
  "Attribute : Canvas.end" -> "Class : Canvas"
  "Attribute : Canvas.figureSelected" -> "Class : Canvas"
  "Attribute : Canvas.color" -> "Class : Canvas"
  "Attribute : Main.toolPanel" -> "Method : Main.initContentPane"
  "Attribute : Main.contentPane" -> "Method : Main.initContentPane"
  "Attribute : Main.canvas" -> "Method : Main.initContentPane"
  "Attribute : Main.contentPane" -> "Method : Main.initLayout"
  "Attribute : Main.toolPanel" -> "Method : Main.initLayout"
  "Attribute : Main.canvas" -> "Method : Main.initListeners"
  "Method : Main.init" -> "Class : Main"
  "Method : Main.initAtoms" -> "Class : Main"
  "Method : Main.initContentPane" -> "Class : Main"
  "Method : Main.initLayout" -> "Class : Main"
  "Method : Main.initListeners" -> "Class : Main"
  "Method : Main.main" -> "Class : Main"
  "Attribute : Main.WIDTH" -> "Class : Main"
  "Attribute : Main.HEIGHT" -> "Class : Main"
  "Attribute : Main.toolPanel" -> "Class : Main"
  "Attribute : Main.canvas" -> "Class : Main"
  "Attribute : Main.contentPane" -> "Class : Main"
}

```

Figure 13: fichier.dot

- 2. A partir du graphe, construisez la matrice de couplage des dépendances structurelles.

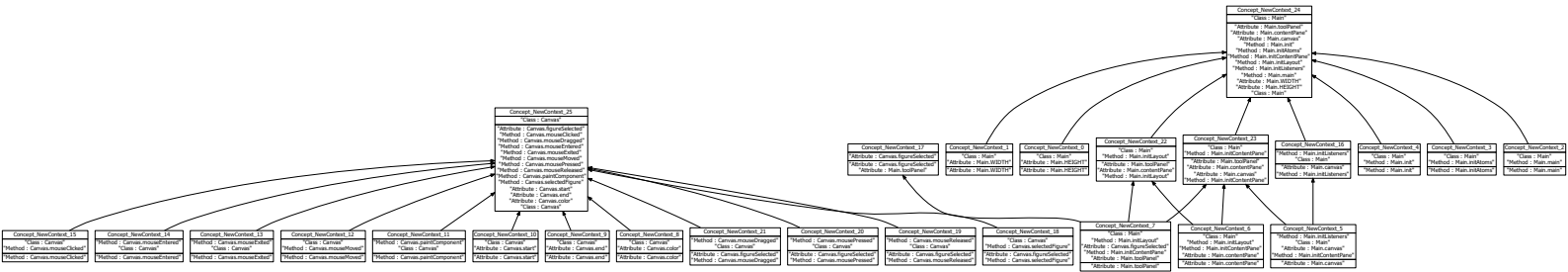
	"Attribute : ..."	"Method : C..."	"Method : C..."	"Method : C..."	"Method : C..."	"Method : C..."	"Method : C..."	"Method : C..."	"Method : C..."	"Method : C..."	"Attribute : ..."	"Attribute : ..."	"Attribute : ..."	"Attribute : ..."	"Attribute : ..."
"Attribute : ..."	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>												
"Method : ..."	<input type="checkbox"/>	<input checked="" type="checkbox"/>						<input checked="" type="checkbox"/>							
"Method : ..."	<input type="checkbox"/>		<input checked="" type="checkbox"/>												
"Method : ..."	<input type="checkbox"/>			<input checked="" type="checkbox"/>											
"Method : ..."	<input type="checkbox"/>				<input checked="" type="checkbox"/>										
"Method : ..."	<input type="checkbox"/>					<input checked="" type="checkbox"/>									
"Method : ..."	<input type="checkbox"/>						<input checked="" type="checkbox"/>								
"Method : ..."	<input type="checkbox"/>							<input checked="" type="checkbox"/>							
"Method : ..."	<input type="checkbox"/>								<input checked="" type="checkbox"/>						
"Attribute : ..."										<input checked="" type="checkbox"/>					
"Attribute : ..."											<input checked="" type="checkbox"/>				
"Attribute : ..."	<input checked="" type="checkbox"/>											<input checked="" type="checkbox"/>			
"Attribute : ..."													<input checked="" type="checkbox"/>		
"Attribute : ..."														<input checked="" type="checkbox"/>	
"Method : ..."															<input checked="" type="checkbox"/>
"Method : ..."															
"Method : ..."															
"Method : ..."															
"Method : ..."															
"Attribute : ..."															
"Attribute : ..."															
"Class : Ma..."															
"Class : Ca..."															

Figure 14: matrice sur rca

## 2.2 Exercice 5 : Les groupes minimaux d'artefacts

Nous avons pendant le cours que prendre aussi en compte le couplage lexical pouvant contribuer à améliorer les résultats. Toutefois, nous allons uniquement considérer le couplage structurel dans ce TP pour découvrir les groupes minimaux d'artefacts.

- 2. Pour ce concept formel, générer avec le treillis de concept à l'aide de rca-explorer, en utilisant ACPOSET comme algorithme à la place de FCA (voir la figure 5).



- 3. Observez-vous un treillis qui apparaît déconnecté ? OUI

Les treillis déconnectés indiquent que vous pouvez extraire l'implémentation de plusieurs caractéristiques pour un même groupe. Ces sous-groupes vous semblent-ils mieux capturer l'implémentation des caractéristiques spécifiques ? OUI