Examen ECRIT - session 1
Documents sur le projet sur l'extraction d'interfaces autorisés

Questions sur le projet d'extraction d'interfaces en Java (*big refactoring*)

Dans cette partie, nous proposons d'appliquer l'approche d'extraction d'interfaces en Java sur les classes permettant de formater des données comme les dates, les messages ou les nombres, notamment avec des informations *locales*, comme l'utilisation, suivant les régions du monde, de la virgule (système anglo-saxon) ou du point (autres systèmes) pour séparer la partie décimale de la partie entière d'un nombre décimal. `Format`, `DateFormat` et `NumberFormat` sont des classes abstraites, les autres classes sont concrètes (`SimpleDateFormat`, `MessageFormat`, `ChoiceFormat`, `DecimalFormat`). Le formatage se fait dans certains cas à l'aide d'un `pattern`, chaîne de caractères décrivant la forme de sortie d'une date, d'un message (concaténation d'informations destinées à être présentées à un utilisateur final), d'un choix (intervalle de nombres), ou enfin d'un nombre. Pour donner des exemples de patterns : `"MM/dd/yyyy"` est un pattern pour une date (01/12/2017 est un formatage pour le 12 janvier 2017), `"0.###E0"` est un formatage pour un nombre décimal en système anglo-saxon (1234 est représenté par "1.234E3").

Le schéma de la figure 1 vous présente l'AOC-poset construit à partir d'une table issue du programme d'extraction que vous avez utilisé pour l'analyse des séquences et qui est appliqué ici aux classes de formatage de données. Le schéma de la figure 2 vous présente le treillis de concepts construit à partir des mêmes données.

L'extension des concepts contient les classes concrètes et abstraites de formatage. Les intensions des concepts contiennent les signatures des méthodes publiques et non statiques de ces classes.

Question 1. (1 point) Les concepts sont représentés dans les deux figures sous forme simplifiée. Donnez l'intension complète (ensemble des signatures) et l'extension complète (ensemble des classes représentées) du concept `Concept_Format_7` de la figure 1.

Question 2. (7 points) Proposez graphiquement une hiérarchie d'interfaces et de classes Java tirée de l'AOC-poset. Indiquez clairement sur votre graphique les relations `extends` (interface-interface ou classe-classe) et `implements` (classe-interface). Ne recopiez pas les méthodes, utilisez les numéros de concepts de la figure 1 pour identifier des blocs de signatures de méthodes et montrer dans quelle(s) interface(s) vous les introduisez. Il n'y a pas une unique solution, donc vous devez préciser la logique qui vous guide. En particulier, donnez des noms aux interfaces de manière à indiquer quelle signification elles véhiculent.

Rappel : en Java, toutes les classes sont directement ou indirectement sous-classes de `Object`. La classe `Format` est directement sous-classe de `Object` dans la hiérarchie d'origine. Les interfaces de Java n'ont pas besoin d'être rattachées sous une racine unique. Une classe ne peut avoir qu'une seule super-classe directe ; une classe peut implémenter directement plusieurs interfaces ; une interface peut spécialiser directement une ou plusieurs interfaces.

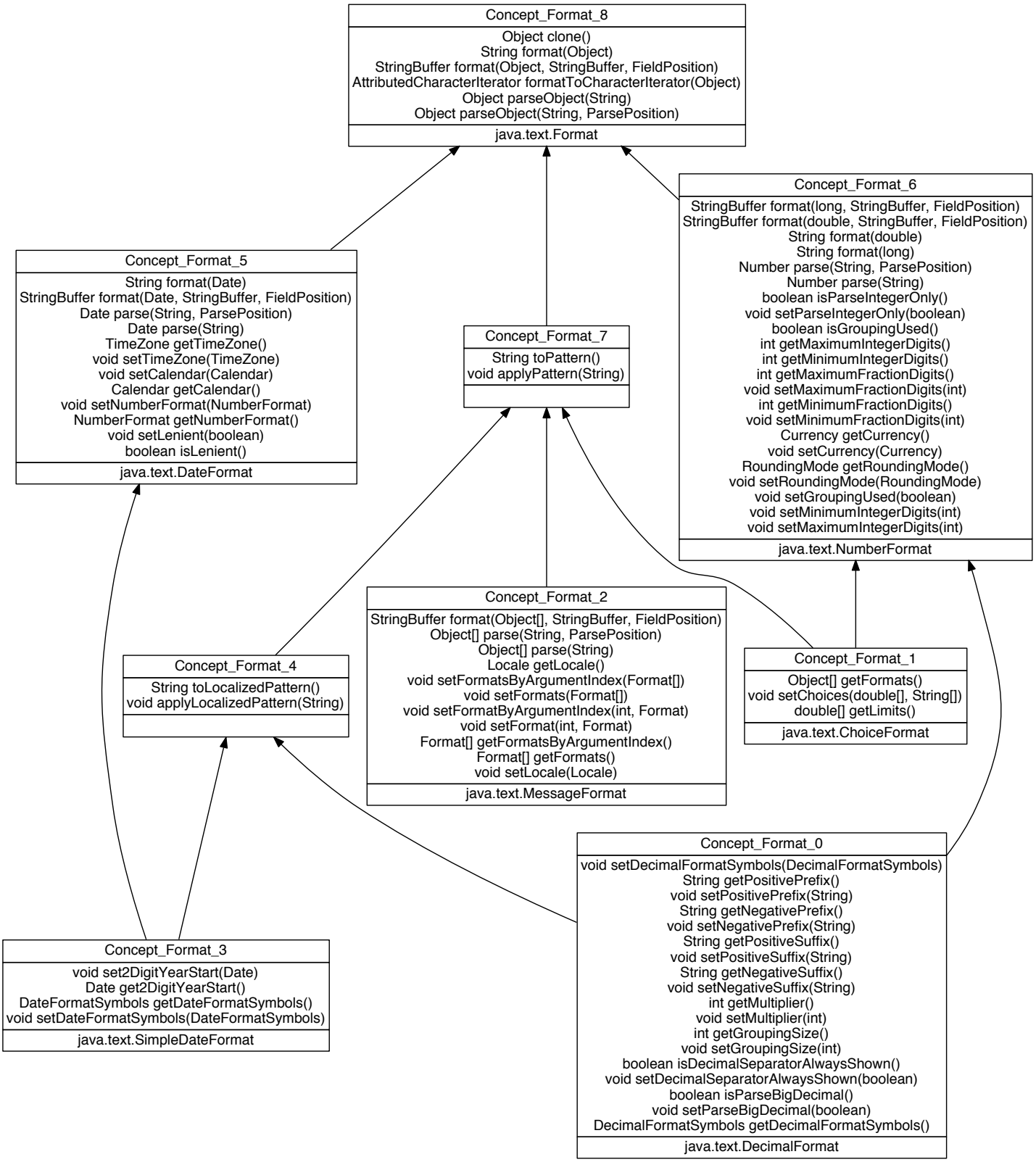

FIGURE 1 – AOC-poset construit sur les signatures des méthodes des classes de formatage

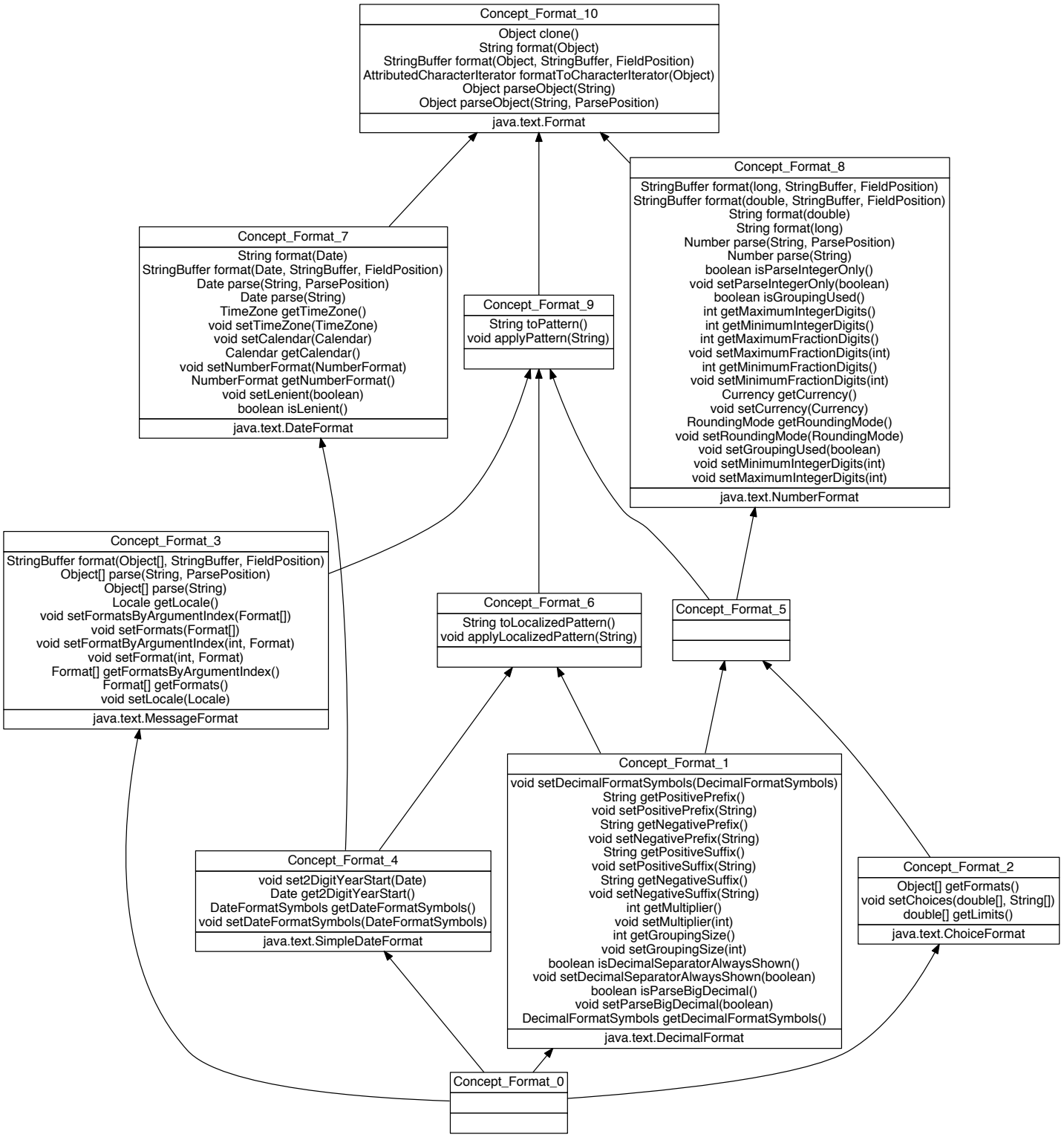

FIGURE 2 – Treillis de concepts construit sur les signatures des méthodes des classes de formatage