

TD n°8 : Pattern Singleton

Le TD comme le TP sur le pattern Singleton se feront en **Java**.

Exercice 1 (Implémentations du Singleton)

Répondre à ces questions sans regarder le cours :

1. On commence soft : faire le diagramme UML du Singleton.
2. Écrire en Java une implémentation du pattern Singleton sans prendre en compte le multithreading.
3. Même chose avec protections pour le multithreading.

Exercice 2 (Singleton Factory Method)

Il est courant de programmer le pattern factory method comme un singleton. Considérons l'exemple suivant : vous développez un système de création de billet. Ce système permet de créer soit des billets papiers, soit des billets électroniques, de manière mutuellement exclusive : si vous choisissez de faire des billets électroniques par exemple, une fois le système instancié vous ne pourrez plus créer de billets papier avec.

On fait appel à une factory pour créer des billets de tombola. Chaque billet à un numéro unique. Il faut éviter d'avoir plusieurs instances de la factory : si deux threads utilisant chacune leur propre instance de factory créent deux billets du même type en même temps, on risque de se retrouver avec des billets papier ou électroniques ayant le même numéro.

1. Faire le diagramme UML pour ce programme.
2. Écrire une première version du programme correspondant à votre diagramme UML.
3. On souhaite faire la modification suivante : maintenant, le même système peut créer à la fois des billets papier ou électroniques une fois instancié, à la demande. On va considérer qu'un billet électronique et un billet papier peuvent porter le même numéro. Comment modifier votre programme ? (UML et code)
4. Et si on veut interdire qu'un billet électronique et un billet papier puisse porter le même numéro ?