



# UML pratique avec RSA Cahier de TP

Copyright - OXIANE, 98 avenue du Gal Leclerc, 92100 Boulogne. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de OXIANE et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de OXIANE.

OXIANE, le logo OXIANE sont des marques de fabrique ou des marques déposées, ou marques de service, de OXIANE en France et dans d'autres pays.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REpondre A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.

Copyright - OXIANE, 98 avenue du Gal Leclerc, 92100 Boulogne. All rights reserved .

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of OXIANE and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from OXIANE suppliers.

OXIANE, the OXIANE logo are trademarks, registered trademarks, or service marks of OXIANE in France and other countries.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

## TP 1 : réfléchir et modéliser avec UML

### Exercice 1 : jeu du nombre secret

Un joueur pense à un nombre compris entre 1 et 100. L'autre doit deviner ce nombre en effectuant des propositions auxquelles on lui répond par « c'est plus », « c'est moins » ou « c'est trouvé ».

Celui qui « pense », comptabilise aussi le nombre de coups à chaque proposition.

- > Recensez les attributs et méthodes de chaque joueur
  - On peut les appeler « JoueurQuiPense » et « JoueurQuiDevine » ou bien « Jeu » et « Joueur »
  - Il peut être utile de réaliser une ou deux parties
- > Utiliser la notation UML objet pour réfléchir sur papier / tableau blanc
- > Utiliser le diagramme de classe dans RSA pour modéliser les classes du jeu du nombre secret

### Exercice 2 : variantes

- > Décortiquez plusieurs variantes de joueur (indiquez les attributs et les messages)
  - Un joueur systématique qui tente toutes les valeurs de 1 à 100
  - Un joueur aléatoire qui tente à chaque fois un nombre au hasard entre 1 et 100
  - Un joueur futé qui ne retente pas un nombre qu'il a déjà proposé
  - Un joueur méthodique qui réduit progressivement l'intervalle des possibilités
  - Un joueur dichotomique qui applique la stratégie du centre

### Exercice 3 : utilisez le diagramme de classes

- > Utilisez la notation UML pour représenter les différentes classes Jeu et Joueur précédentes
  - Indiquez notamment les méthodes privées et publiques
- > Utilisez les notes pour expliquer les différentes stratégies de jeu de l'exercice précédent
  - Vous pouvez aussi utiliser les notes pour indiquer un pseudo code des méthodes

### Exercice 4 (optionnel) : utilisez le diagramme d'objets

- > Utiliser le diagramme d'objet pour illustrer une partie avec le joueur dichotomique

## TP2 : utiliser le diagramme de séquence

### Exercice 1 : utilisez le diagramme de séquence (ou diagramme d'interactions)

- > Représentez la séquence de messages échangés entre le joueur et le jeu
  - Réaliser plusieurs diagrammes pour illustrer les parties « coup de bol » et « en deux coups »
  - Utilisez les fragments (boucle, alternative, ...) pour expliquer le déroulement de la partie générale

### **Exercice 2 : commencez l'analyse d'un cas plus complexe (l'application CaveAVin)**

- > Prenez connaissance du cahier des charges de l'application (situé en fin de cahier de tp)
  - Lisez attentivement et essayez de comprendre le système et sa finalité
- > Identifiez quelques classes
  - Repérez les noms et verbes (ils se traduisent en général par des classes et des méthodes)
- > Proposez un premier diagramme de classes (dans un nouveau projet)
  - Représentez les classes identifiées avec leurs attributs et quelques méthodes
  - Faites attention au choix des noms (choisir un bon nom prend souvent beaucoup de temps)
  - Il n'est généralement pas nécessaire de préciser les types pendant cette phase de l'analyse
- > Confrontez votre travail avec vos collègues
  - UML sert à communiquer et à présenter, défendre et argumenter une modélisation
  - Il n'y a pas qu'une seule modélisation possible
  - Il y a des modélisations plus ou moins utiles, précises et évolutives
  - L'élaboration d'un modèle est un travail itératif et incrémental

## TP 3 : détailler les éléments UML

### **Exercice 1 : détaillez les classes vues précédemment**

- > Préciser les types, les paramètres, les valeurs par défaut, ...
  - Ajouter des constantes (statique) pour indiquer les réponses possibles du jeu du nombre secret
- > Détailler l'analyse de la cave à vin

## TP 4 : représenter les relations

### **Exercice 1 : ajouter les relations dans le modèle CaveAVin**

- > Ajoutez les associations
  - Faites attention au choix du type d'association (association simple, agrégation ou composition)
  - Réfléchissez et indiquez les multiplicités et éventuellement le sens de navigation
  - Ce travail vous amènera peut-être à créer de nouvelles classes ou à en transformer

## TP 5 : héritage et autres relations

### **Exercice 1 : utilisez les relations d'héritage et les autres relations**

- > Généralisez (ou spécialisez) pour faire émerger des liens d'héritage
- > Introduisez une classe représentant le générateur de nombres aléatoires dans le jeu
  - Utilisez la relation de dépendance pour montrer l'utilisation de cette classe

- Détaillez les diagrammes de séquence pour montrer les échanges de messages

## TP 6 : interfaces, stéréotypes et contraintes

### *Exercice 1 : revisitez la conception du jeu du nombre secret*

- > Réalisez une nouvelle conception du jeu basée sur les interfaces
  - Introduisez une énumération **Reponse** pour les retours possibles du jeu
  - Introduisez des contraintes

## TP 7 : diagramme d'états

### *Exercice 1 : utilisez les diagrammes d'états*

- > Décrivez les états et transitions possibles du jeu du nombre secret
- > Ajoutez la possibilité de rejouer et d'abandonner une partie
- > Utilisez le diagramme d'état pour montrer les différents états possibles de la commande de vin
  - Étudiez d'abord le cas nominal puis ajoutez les situations anormales (annulations, ...)

## TP 8 : diagramme de cas d'utilisation

### *Exercice 1 : identifier les acteurs*

- > Recensez les acteurs de la cave à vin
  - Identifiez et différenciez les acteurs principaux et secondaires, humains et « système externe »
- > Recensez les cas d'utilisation principaux de la cave à vin
  - Utilisez un cadre pour délimiter les frontières du système
  - Connectez les acteurs (principaux à gauche, secondaires à droite)
- > Explicitez les interactions de quelques cas d'utilisation principaux
  - Utilisez le diagramme d'interactions (séquences)
  - Pensez à représenter le système comme un acteur dans ces diagrammes
  - Pensez à rester à un niveau général (ne pas détailler ce qu'il se passe dans le système)
- > Ajoutez les cas d'utilisations secondaires
  - Réorganisez et utilisez les différents liens tout en gardant un diagramme simple et lisible

## TP 9 : Diagramme d'activités

### *Exercice 1 : flot d'actions et flot de données*

- > Utilisez le diagramme d'activités pour expliciter le processus général de la commande

## Cave à vin

On souhaite réaliser un logiciel de gestion de cave à vins à destination des particuliers.

L'utilisateur pourra gérer ses bouteilles dans sa cave.

Une bouteille est identifiée par son nom (par exemple « Château Margaux »), sa région (par exemple « Bordeaux »), son appellation (par exemple « Medoc »), sa couleur (Rouge, Rosé, Blanc), son millésime (l'année), le viticulteur (ou producteur).

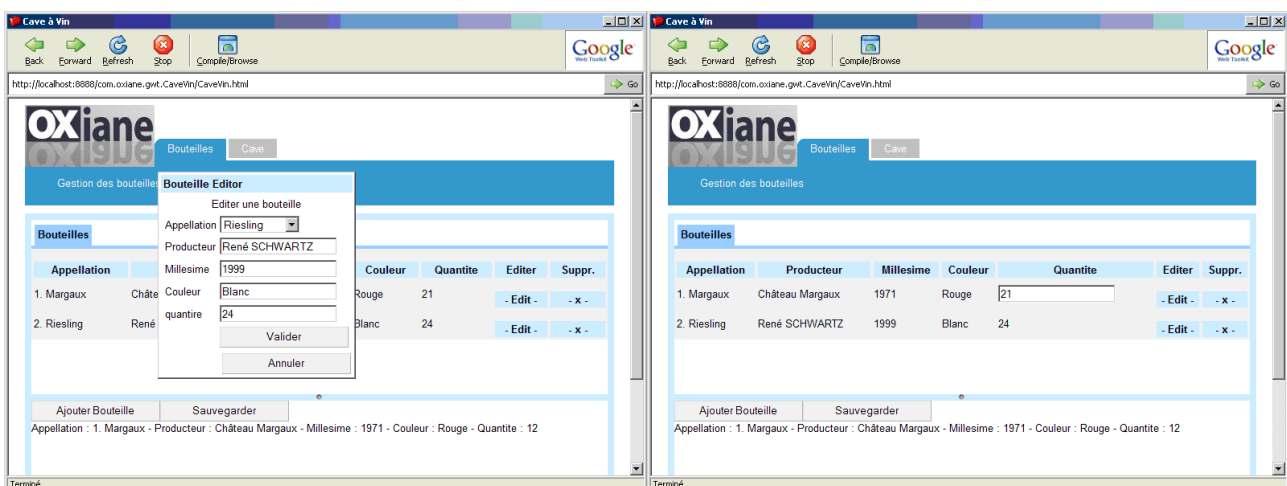
L'utilisateur veut pouvoir enregistrer le nombre de bouteilles dont il dispose dans sa cave.

Par exemple :

- 12 Saint Emilion Château Ausone de 2002 (un Bordeaux rouge)
- 24 Muscadet ...
- etc.

Il doit pouvoir enregistrer de nouvelles entrées, retirer des bouteilles bues, etc.

On souhaite une application en mode web.



On souhaite également rendre l'application de cave à vins disponible aux viticulteurs. Il pourra ainsi renseigner l'état de ses stocks.

L'utilisateur qui gère sa cave à vins aura accès aux producteurs et à ses stocks disponibles, pourra rechercher les bouteilles de différentes manières, trouver le producteur associé et passer commande.

Le site fera office d'intermédiaire. Tout mode de paiement doit être possible (chèque, virement, paiement en ligne).

Le site prélève une commission sur les transactions.

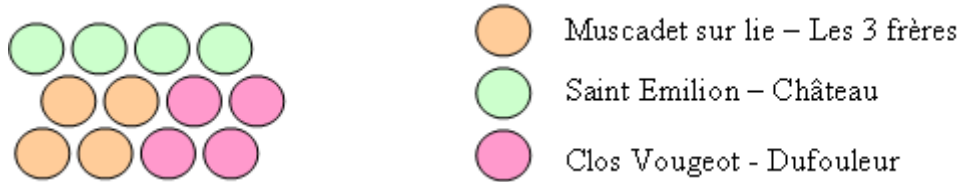
Lorsqu'une commande est passée, le producteur en est averti. Il peut voir l'état des commandes en cours et leur statut.

Lorsqu'un règlement est reçu, le site avertit le producteur qui peut passer à l'envoi. Le site reversera le montant au producteur après prélèvement de sa commission.

L'utilisateur peut suivre ses commandes et leur statut (attente de règlement, règlement reçu, assemblage, expédiée).

Une commande expédiée pourra être associée à un numéro (pour le moment, on ne supportera que colissimo), ce qui permettra à l'utilisateur de disposer d'un lien vers le suivi de sa commande colissimo.

Du côté de la gestion de cave à vins pour un utilisateur, on souhaite ajouter une fonctionnalité permettant de visualiser sur un schéma 2D la disposition des bouteilles :



Enfin, la société devra pouvoir avoir accès à toutes les données : utilisateurs, viticulteurs, commandes en cours, etc.