

Projets de programmation orientée objets

Proposition de sujets de Projet en JAVA Ce document comporte 5 projets. Chaque binôme doit choisir un projet. Il est possible d'avoir un trinôme par projet à condition que la réalisation soit à la hauteur en intégrant de nouvelles fonctionnalités ou encore une interface graphique. Il est à noter que toute initiative de rajouter ou encore de complexifier le projet est très appréciée. Validez vos propositions avec le chargé de TP. Le projet se déroule sur quatre étapes :

- **Étape 1** : consiste à identifier les classes (attributs et méthodes) ainsi que les liens entre eux (utilisation, composition ou encore héritage). Une fois la liste des classes identifiée, validez votre solution ainsi que vos choix avec le chargé de TP (Diagramme de classes UML).
- **Étape 2** : Programmation et préparation des tests.
- **Étape 3** : Rédaction du rapport. Il s'agit de quelques pages (3 maximum) décrivant le projet, les fonctionnalités ainsi que le modèle de donnée de l'étape 1.
- **Étape 4** : Soutenance.

Projet 1 : Développer un Système de Gestion de Citoyens [O.O+ 1 PERSONNE, VOIR PLUS SI AJOUT](#)

Le logiciel doit permettre à une mairie de réaliser la gestion de ses citoyens. Une personne est caractérisée par son N° d'identifiant unique, son nom, son prénom, sa date de naissance, son sexe, son état civil (mariée, divorcée, célibataire ou veuve), et le N° d'identifiant de son conjoint (si elle est mariée).

Objectifs : Créer un menu permettant de choisir parmi les fonctions suivantes : Mariage d'une personne, divorce d'une personne, connaissance de l'état d'une personne, affichage de la liste de toutes les personnes. Cela se résume par le développement de fonctions de mariage, de divorce, de recherche et d'affichage ainsi que le bon déroulement de chacune de ses fonctions.

Le programme principal offre un menu à l'écran qui permet de :

1. Mariage
2. Divorce
3. Naissance
4. État d'une personne
5. Affichage de la liste des personnes
6. Saisie des personnes
7. Quitter le programme

Lors du choix 1, la fonction demande le n° d'identification des personnes à marier. Si une des personnes est déjà mariée, le programme indique qu'elle est déjà mariée, sinon les personnes sont marquées comme étant mariées.

Lors du choix 2, la fonction demande le n° d'identification de la personne à divorcer. Si la personne n'existe pas, le programme signale l'erreur ; si la personne n'était pas mariée, le programme le stipule sinon la personne est marquée comme étant divorcée.

Lors du choix 3, la fonction demande le n° d'identification des parents et rajoute un nouveau née.

Lors du choix 4, la fonction demande le N° d'identification de la personne dont on désire connaître l'état. Si la personne n'existe pas dans la liste de la mairie une erreur est signalée, sinon son nom, son prénom, son sexe, sa date de naissance et le nom et prénom de son conjoint éventuel sont affichés.

Lors du choix 5, la fonction affiche une liste de l'ensemble des personnes enregistrées à la mairie indiquant : leurs noms, prénoms, sexes, dates de naissance et le nom et prénom du conjoint éventuel.

Lors du choix 6, permet de saisir une personne

Lors du choix 7, la fonction permet de quitter le programme et sauvegarder les données

Projet 2 : Les tours de Hanoi [O.O- 1 PERSONNE](#)

Les tours de Hanoi Les tours de Hanoi est un jeu composé de trois poteaux et N disques de tailles différentes (voir figure). Initialement les disques sont empilés dans l'ordre décroissant sur un poteau qu'on appelle source. L'objectif du jeu consiste à opérer des déplacements afin de bouger tous les disques vers un des deux autres poteaux (poteau destination). Le déplacement consiste à dépiler un disque au sommet d'un poteau et l'empiler sur un autre poteau, à condition que la taille du disque à son sommet est plus grand que le disque déplacé. Nous voulons réaliser un programme qui permet à l'utilisateur de jouer à ce jeu. Initialement le joueur décide du nombre de disques. Ensuite, le programme propose, soit la résolution automatique du problème soit une résolution manuelle. La résolution automatique affiche la liste des instructions à faire pour résoudre le problème. La résolution manuelle permet à l'utilisateur d'entrer des instructions de déplacement. À tout moment le programme permet d'afficher l'état du jeu ou encore d'annuler en résolvant automatiquement le jeu à partir de l'état courant. Une fois le jeu gagné le programme affiche si le joueur a résolu de manière optimal ou pas. Si c'est pas le cas, le programme affiche les déplacements inutiles p/r à la solution optimal.

Projet 3 : Location de scooter [O.O 2 PERSONNES PROPRIETAIRE DU PARC](#)

Le logiciel permet au magasin « LOUSCOOT », commerce de location de scooters de réaliser la gestion de son parc. Un scooter est caractérisé par son modèle, son numéro d'identification, son kilométrage et son état (disponible ou en cours de location).

Objectifs : Créer un menu permettant de choisir parmi les fonctions suivantes : Location d'un scooter, réception d'un scooter, connaissance de l'état d'un scooter, affichage du parc de

scooters. Cela se résume par le développement de fonctions de location, de réception, de recherche et d'affichage ainsi que le bon déroulement de chacune de ses fonctions. Le programme principal offre un menu à l'écran qui permet de :

1. Louer un scooter
2. Retour d'un scooter
3. État d'un scooter
4. Affichage de l'état du parc de scooters
5. Saisie du parc des scooters
6. Quitter le programme

Après l'exécution de ces cinq options, le programme réaffiche le menu

Lors du choix 1, la fonction demande le n° d'identification du scooter à louer. Si le scooter n'existe pas, le programme signale l'erreur ; s'il est déjà loué, le programme indique qu'il est déjà en location, sinon le scooter est marqué comme étant en location.

Lors du choix 2, la fonction demande le n° d'identification du scooter à retourner. Si le scooter n'existe pas, le programme signale l'erreur ; si le scooter n'était pas marqué comme étant en location, le programme le stipule sinon le programme demande le nombre de kilomètre effectué et les rajoute au kilométrage du scooter. Le scooter est alors marqué comme étant disponible.

Lors du choix 3, la fonction demande le N° d'identification du scooter dont on désire connaître l'état. Si le scooter n'existe pas dans le parc une erreur est signalée, sinon son modèle, son N° d'identification, son kilométrage et son état de location sont affichés.

Lors du choix 4, la fonction affiche un état de l'ensemble du parc de scooters, c'est-à-dire la liste indiquant : le N° d'identification des scooters, le modèle du scooter, le kilométrage du scooter, et son état.

Lors du choix 5, la fonction affiche un état résumé de l'ensemble du parc de scooters, c'est à dire :

- Le Nombre total de scooters
- Le Nombre de scooters en location et leur N° d'identification,
- Le Nombre de scooters disponibles et leur N° d'identification,
- Le kilométrage moyen de l'ensemble des scooters

Lors du choix 6, il y a sauvegarde les informations contenues dans un fichier texte. Au lancement du programme, il y a restauration de l'ensemble des informations contenues dans le fichier de sauvegarde.

Projet 4 livraison de pizza [O.O 2 PERSONNES](#)

On veut modéliser la gestion d'une entreprise de fabrication et de livraison de pizzas à domicile: la société RaPizz. Le client choisit sa pizza à partir d'un catalogue. Une pizza est caractérisée par son nom, les ingrédients qui la composent et son prix de base. Pour chaque pizza, il existe 3 tailles : "naine", "humaine" et "ogresse". La naine est 1/3 moins chère que le prix de base, la pizza "humaine" coûte le prix de base et l' "ogresse" est 1/3 plus chère que le prix de base. Les pizzas sont livrées par des livreurs qui circulent en voiture ou à moto. Pour passer une commande de pizza, le client fournit son numéro de téléphone (qui va permettre de l'identifier)

et le contenu de sa commande qu'on représente sous la forme d'une liste composée du nom d'une pizza, de sa taille et la quantité commandée.

Le système commence par vérifier que le client existe dans la base des clients et conserve l'objet client qui lui correspond. Le système vérifie ensuite que la commande est faisable (elle peut être fabriquée dans un Point RaPizza qui fait toutes les pizzas spéciales demandées) puis crée la commande dans le système et en indique le numéro et le prix au client. Le mode de paiement est du type prépayé c'est-à-dire que le client est abonné au service de livraison de pizza et doit approvisionner son compte. On vérifie le solde du client avant de préparer et de livrer les pizzas. Une pizza gratuite sera livrée au bout de 10 pizzas achetées. Toute pizza livrée en plus de 30 minutes est gratuite (retard de livraison). A partir de cette description du cahier des charges "Livraison de commande de pizzas", on vous demande dans un premier temps d'élaborer le logiciel qui permet de l'automatiser.

projet 5 : Arbre généalogique [O.O- 1 PERSONNE](#)

Pour ce projet, nous nous intéresserons à la manipulation d'arbres généalogiques. Un arbre généalogique est représenté par un assemblage d'objet de type personne. Chaque objet contient une référence vers le prénom d'une personne ainsi que des références vers son père, sa mère et ses enfants. Une personne est identifiée par son prénom, son sexe, sa date de naissance et de mort (null si vivant). Le programme doit permettre de saisir un arbre généalogique. Ensuite il offre le menu suivant :

1. Afficher l'arbre
2. Afficher les ancêtres d'une personne donnée
3. Afficher les (demi) frères et (demi) sœurs d'une personne donnée
4. Afficher les cousins d'une personne donnée
5. Donner le lien de parenté entre deux personnes données

La dernière question constitue la partie ouverte du projet. Il faut trouver un moyen qui permet de préciser de manière systématique le lien qui relie deux personnes.

Projet 6 : Gestion d'un magasin de vente [O.O 1 PERSONNE SI AJOUT 2 PERSONNES](#)

On souhaite mettre en place un outil de gestion d'un magasin de vente d'article informatique.

Cet outil doit intégrer les fonctionnalités suivantes :

Gestion de produits : ajout/suppression d'articles

Gestion de client : ajout/suppression d'articles

Vente : effectuer des ventes pour les clients

Statistique : visualisation des statistiques de vente par article ou par client

Ce projet semble simple mais il constitue un vrai défi de conception. Il est ouvert et donne aux étudiants qui le choisissent une vraie marge de complexification.

Projet 7 : gestion d'hôtels [O.O+ 2 PERSONNES VOIR 3](#)

Un hôtel comprend quatre types de chambres : double, simple, suite normale et suite présidentielle. Chaque chambre possède un numéro unique et l'étage. Les clients de l'hôtel peuvent effectuer des réservations. Une réservation est définie par la date de début, date de fin ainsi que le client concerné. Le client peut soit annuler la réservation soit séjourner. Dans le cas

où le client n'annule pas, la réservation donne lieu à un séjour. En plus du coût de la chambre, le client peut consommer durant son séjour des produits au mini bar. On voudrait écrire un petit logiciel de réservation pour cet hôtel pour une durée d'une année. Le programme offre un menu principal qui permet :

1. La gestion des chambres (saisies, modification)
2. La gestion des clients
3. La gestion des réservations :
 - a. Chercher des chambres disponibles pour une période donnée
 - b. Ajouter une réservation
 - c. Annuler une réservation
4. La gestion des séjours :
 - a. Créer un séjour
 - b. Ajouter des consommations mini bar
 - c. Facturation

Ce projet semble simple mais il constitue un vrai défi de conception. Il est ouvert et donne aux étudiants qui le choisissent une vraie marge de complexification.