

Examen intra

## Identification du cours

Cours 420-2P6-VL- Programmation orientée objet

Professeure : Farida Mostefaoui

Session : Hiver 2022

## Pondération

Cet examen vaut **15%** de la note finale

## Date & heure

**L’examen est disponible le lundi 21 mars de 8h à 00h**.

L’examen est d'une durée estimée à 2 heures.

Remise obligatoire par Léa, au plus tard à le lundi 21 mars à 00h00.

Aucun retard ne sera accepté.

## Consignes

Toute documentation est permise.

L’examen doit être fait individuellement.

Attention au **PLAGIAT**. Les programmes identiques recevront la note 0.

L’environnement Eclipse doit être utilisé.

## Critères d’évaluation

* Programme correct, lisible et compréhensible.
* Exécution correcte.
* Respect des indications données.
* Qualité du code
* Orthographe correcte des chaînes (Pénalité jusqu’à 10%).

## Modalité de remise

Remise obligatoire par Léa, au plus tard à 00h00.

Aucun retard ne sera accepté.

Exporter le projet par Eclipse et le remettre par Léa.

Vérifier que votre remise est correcte.

## Critères d’évaluation

1. Respect des normes et conventions d’écriture des programmes.
2. Respect des noms donnés pour les classes, attributs et méthodes.
3. Compilation et exécution correctes.
4. Fonctionnalités correctes des méthodes.
5. Jeux d’essais pertinents.
6. Qualité du code et commentaire.

## Énoncé

On s’intéresse à la gestion d’une compagnie d’énergie qui facture ses clients selon le modèle de tarification progressive, plus on consomme plus cher on paye. Il y a trois paliers de facturation, selon le nombre de  kWh consommés mensuellement (pour des raisons de simplicité, on ne considère que le traitement mensuel):

Palier 1 : 1 à 500, le prix est de 5 cents le kWh,

Palier 2 : 501 à 1000, le prix est de 7 cents le kWh,

Palier 3 : Plus que 1000, le prix est de 10 cents le kWh.

Les consommations énergétiques sont exprimées en heures entières.

# Travail demandé

* Utiliser les modificateurs de visibilité : private pour les attributs et public pour les méthodes.
* Écrire les getters et les setters au besoin (seulement ceux utilisés).
* Utiliser le mot clé static quand nécessaire.
* Respecter les noms de classes, d’attributs et de méthodes donnés.

1. Écrire la classe Client qui modélise les clients et quicontient :

* Le nom du client nom (String).
* Le prénom du client prénom (String).
* L’adresse du client adresse (String).
* Le nombre total de kWh consommés mensuellement kWh(int).

Le constructeur Client qui crée les objets de type Client et initialise ses attributs.

* La méthode getPalier() qui permet de déterminer et de retourner le palier de consommation (int) du client et retourne le palier.
* La méthode calculerTotal() qui permet de calculer et de retourner le prix total en $ à payer (float) pour un client. Cette méthode utilise la méthode getPalier().
* La méthode toString()qui retourne les attributs d’un client sous forme de chaine.
* La méthode comparer(Client c1, Client c2) qui prend en paramètres deux clients c1 et c2 et qui retourne le client qui a la plus grande consommation.

Attention cette méthode doit être statique, car elle ne décrit pas un objet particulier.

1. Décrire la classe Listequicontient la liste des clients. Cette classe est décrite par :

* La liste liste des clients représentée par un tableau de clients. On considère que le nombre maximum de clients est de 500 au plus.
* Le constructeur de la classe Liste.

Note : à la création de la liste, celle-ci ne contient aucun client.

* La méthode rajouterClient(Client c) qui insère le client cà la fin de liste si celle-ci n’est pas pleine. Cette méthode retourne true si l’insertion a réussi, false sinon.
* La méthode rechercherPalier(palier p) qui compte et retourne le nombre de clients du palier p qu’il y a dans liste.
* La méthode consommationMax() qui recherche dans liste le client ayant la plus grande consommation de kWh. Ce client est retourné. Si plus d’un client est trouvé, il suffit de retourner un.

1. Écrire la classe Main qui contient la méthode main().

Écrire en codage dur (hardcoding(rentrer les information), sans saisie à l’écran) dans la méthode main()le code Java qui réalise ce qui suit :

1. Créer 6 clients ayant des consommations de paliers différents et les ajouter au fur et à mesure dans la liste des clients.

À la création des objets, choisir des valeurs adéquates pour les différents attributs.

1. Parcourir la liste des clients et afficher pour chaque client ses attributs, son pallier et son total de consommation.
2. Afficher le nombre de clients de chaque palier.
3. Parmi les clients créés, afficher le nom et l’adresse du client qui a la plus grande consommation.

Bon travail.