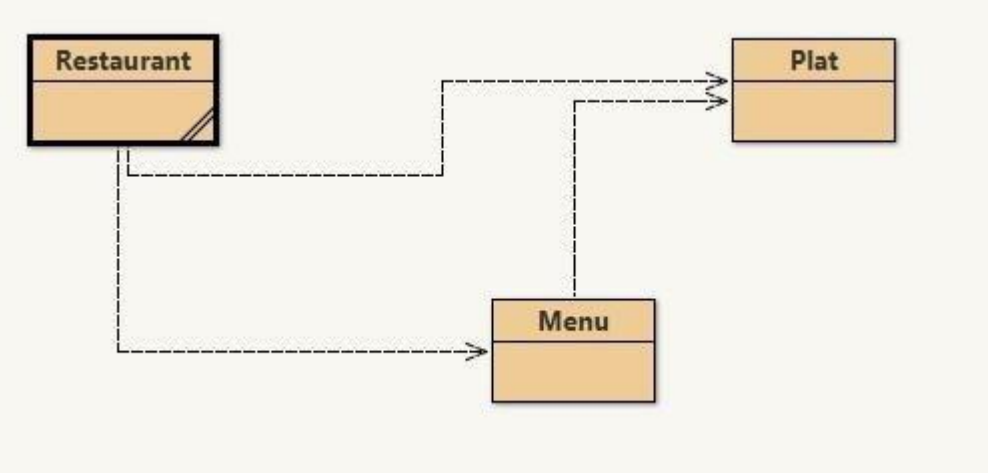
TP03

1. Restaurant



La classe Plat représente un plat servit dans le restaurant et plusieurs plats composent un menu. Un plat comporte deux arguments à savoir : nom et typePlat. Ce dernier permet de savoir si c'est une entrée, un plat, un dessert ou autre. Un plat est identifié par son nom et son type.

La classe Menu représente un ensemble de plats mis ensemble afin de pouvoir offrir une offre spéciale au client. Un menu est caractérisé par un nom, un prix et une liste de plats et identifié par son nom.

Cette classe doit comporter une méthode permettant d'ajouter un plat à la liste de plat. Cependant, un menu ne peut excéder une taille de 5 plats. Un message d'erreur est affiché lorsque l'on essaie d'en ajouter en trop (exemple de sortie).

La classe Restaurant sera votre classe contenant la main() et c'est dans celle-ci que vous effectuerez les créations d'instances et appels de méthodes (La classe Test).

Dans cette classe se trouvera une ArrayList des menus. Une méthode permettra de rechercher un plat particulier parmi les menus. Attentions aux responsabilités !

Exemples de sorties :

```
Erreur : Il ne peut pas y avoir plus que 5 plats dans un menu.


Recherche de Cassoulet (Plat)
    Le plat Cassoulet (Plat) se trouve dans le Menu du jour à 35Chf composé de :
    Cassoulet (Plat) Profiteroles (Dessert)

    Le plat Cassoulet (Plat) se trouve dans le Menu de saison à 40Chf composé de :
    Salade Niçoise (Entrée/Plat) Cassoulet (Plat) Profiteroles (Dessert)

    Le plat Cassoulet (Plat) se trouve dans le Menu duo à 60Chf composé de :
    Cassoulet (Plat) Pâtes au saumon (Plat) Salade verte (Entrée) Assiette Valaisanne (Entrée/Plat) Coupe Danemark (Dessert)

    Le plat Cassoulet (Plat) se trouve dans le Menu enfant à 25Chf composé de :
    Cassoulet (Plat) Pâtes au saumon (Plat)


Recherche de Quiche Lorraine (Plat)
    Le plat Quiche Lorraine (Plat) n'est dans aucun menu.

Process finished with exit code 0
```

2. Courses de ski

Créez une classe **Resultat** dans le package **Domaine** contenant les résultats d'une petite course amateur de ski. Les 2 seules informations à gérer sont le numéro de dossard, ainsi que le temps effectué *(en seconde et dixièmes)*. Chaque skieur *(représenté par son dossard)* peut effectuer plusieurs fois la course, il pourrait donc y avoir plusieurs instances *(plusieurs fois le même numéro de dossard)* dans la liste des résultats.

Utilisez l'application (classe) **StatCourse** qui affiche différentes statistiques sur les résultats de la course. Voiciles différentes procédures/fonctions que vous devez créer :

- **chargerResultats** qui récupère les résultats de la course, puis retourne **une liste** contenant tous ces résultats. Deux fonctions se trouvant dans la classe **Bdd** permettent de récupérer les informations :
    - **public static int[] recupererLesDossards()** retourne un tableau contenant les dossards dans l'ordre de passage de l'arrivée
    - **private static double[] recupererLesTemps()** retourne un tableau contenant tous les temps (dans le même ordre que les dossards; par conséquent, tabDossards[2] contiendra le numéro du dossard du 3ème skieur, et tabTemps[2] contiendra son temps).

Grâce à cette fonction **chargerResultats**, les autres méthodes pourront travailler sur une liste de résultats et non pas sur les différents tableaux de dossards, de temps, ….

- **afficherResultats** qui affiche le tableau des résultats reçu en paramètre
- **plusRapide** qui retourne le meilleur résultat de la course
- **nbFois** qui indique combien de fois un certain skieur *(dossard)* a effectué la course *(combien de fois apparait ce dossard dans la liste).* Dans le cas où un skieur n'a pas effectué la course, il faut le préciser avec un message spécifique.
- **afficherStats** : méthode qui appelle toutes les autres ci-dessus.

Exemple de sorties produites

Dans le fichier Main : *Application.StatCourse.afficherStats();*

11 résultats :

Dossard	102 :	59.0
Dossard	111 :	64.0
Dossard	113 :	45.7
Dossard	103 :	54.9
Dossard	116 :	52.4
Dossard	115 :	60.4
Dossard	107 :	68.8
Dossard	116 :	44.2
Dossard	108 :	56.6
Dossard	104 :	63.3
Dossard	112 :	61.2

Plus rapide : Dossard 116 : 44.2

Le skieur au dossard '116' a effectué 2 fois la course

Prenez le temps de bien comprendre l'ordre des opérations avant de commencer à coder, je vous suggère aussi de faire votre propre schéma afin de mieux vous repérer !!!