

Durant cette séance, la majorité des groupes aura son premier Contrôle Continu de TD.

Il sera court, portant sur des questions de cours, les modalités peuvent varier légèrement d'un groupe à l'autre.

Le reste du temps de la séance sera occupé par le début de ce TD. Idéalement ce serait bien de terminer ensemble la section *Définition de la file d'attente*.

## Modélisation d'un site de vente

L'objectif général est de réaliser un site de vente sur internet qui propose des articles uniques. Lorsque plusieurs clients convoitent le même article, le site met en place une file d'attente virtuelle qui fonctionne sur le modèle du "premier arrivé, premier servi". Le premier client qui ajoute l'article dans son panier se place en tête. Les clients suivants qui font de même sont placés selon leur ordre d'arrivée. Si un client renonce au produit (c.à.d retire l'article de son panier), il est aussi retiré de la file d'attente correspondante.

## Définition des clients

On définit une classe `Client` qui aura comme attributs **privés** :

- un attribut `nom` de type chaîne de caractères,
- un attribut `prenom` de type chaîne de caractères.

Écrivez cette classe en la munissant d'un constructeur. On veillera à ce que `nom` et `prenom` ne puissent pas être modifiés, et on écrira des méthodes `getNom` et `getPrenom` pour y accéder.

## Définition de la file d'attente

Les listes, piles et files seront étudiées un peu plus tard en détail dans le cours. On vous propose ici de réaliser une implémentation basée sur les tableaux. Nous la ferons pas à pas.

1. Définissez une classe `FileDAttente` munie d'un attribut `clientsPossibles` (un tableau de clients) destiné à représenter la file d'attente de chaque produit.
2. ajoutez un attribut entier `premierePlaceLibre` qui permettra de savoir où viendra se placer un client qui souhaite rentrer dans la file d'attente.
3. Ajoutez un constructeur sans argument qui initialise une file d'attente où pourront se placer jusqu'à 10 clients.
4. Ajoutez une méthode `getTaille()` qui renvoie le nombre de clients qu'elle contient.
5. Pour insérer un client, on passera par la méthode `boolean ajouterClient(Client x)`. Celle-ci doit :
  - prendre garde d'éviter les doublons,
  - ne pas l'ajouter s'il n'y a pas de place
  - répondre `true` ssi l'insertion s'est bien passée
6. Écrivez une méthode `void desiste(Client x)` qui retire un client de la file d'attente. Il y a un travail à faire : tous les clients qui étaient situés après `x` vont donc avancer d'une position

dans la liste d'attente ; la liste contiendra un client de moins (s'il était bien présent dans la liste)

7. Écrivez une méthode `Client extraitPremier()` qui sort de la file celui qui est prioritaire. Que faut-il retourner si la file d'attente est vide ?
8. Écrivez une méthode `afficher` qui affiche les clients en précisant leur ordre d'attente, par exemple :
  1. Agnès
  2. Olivier
  3. Didier

## Définition des articles

1. Écrivez une classe `Article` avec comme attributs :
  - `description` de type chaîne de caractères, invariable
  - `file` de type `FileDAttente`,
  - `identifiant` de type entier.Le champs `file` représente tous ceux qui attendent cet article. L'attribut `identifiant` est un numéro qui doit être unique pour ce produit. On écrira un constructeur qui prend en paramètre la désignation de l'article et on initialisera le reste des attributs.
2. Écrivez une méthode `estChoisiPar` qui prend un objet `Client` en paramètre et le positionne pour l'achat de cet article.

## Gestion d'un stock et d'une archive (facultatif)

On s'intéresse ici à la gestion de l'ensemble des articles d'un site. Modélisez un site par un objet caractérisé par son adresse postale, et par deux tableaux d'articles : un qui exprime le stock de ceux qui sont en vente, et l'autre qui servira à archiver ceux qui ont été vendus.

Imaginez et implémentez les opérations de gestion qui seraient utiles.