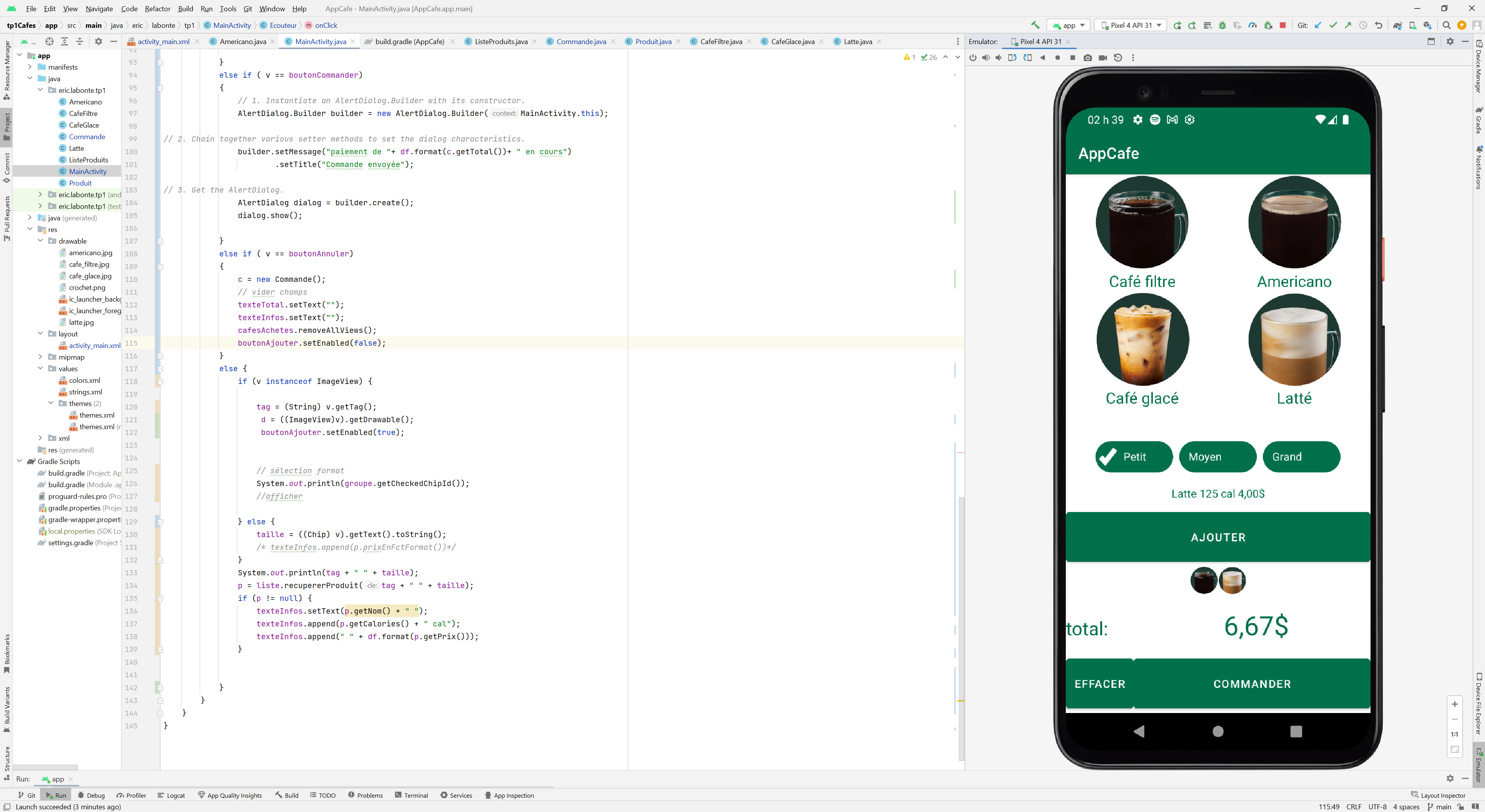
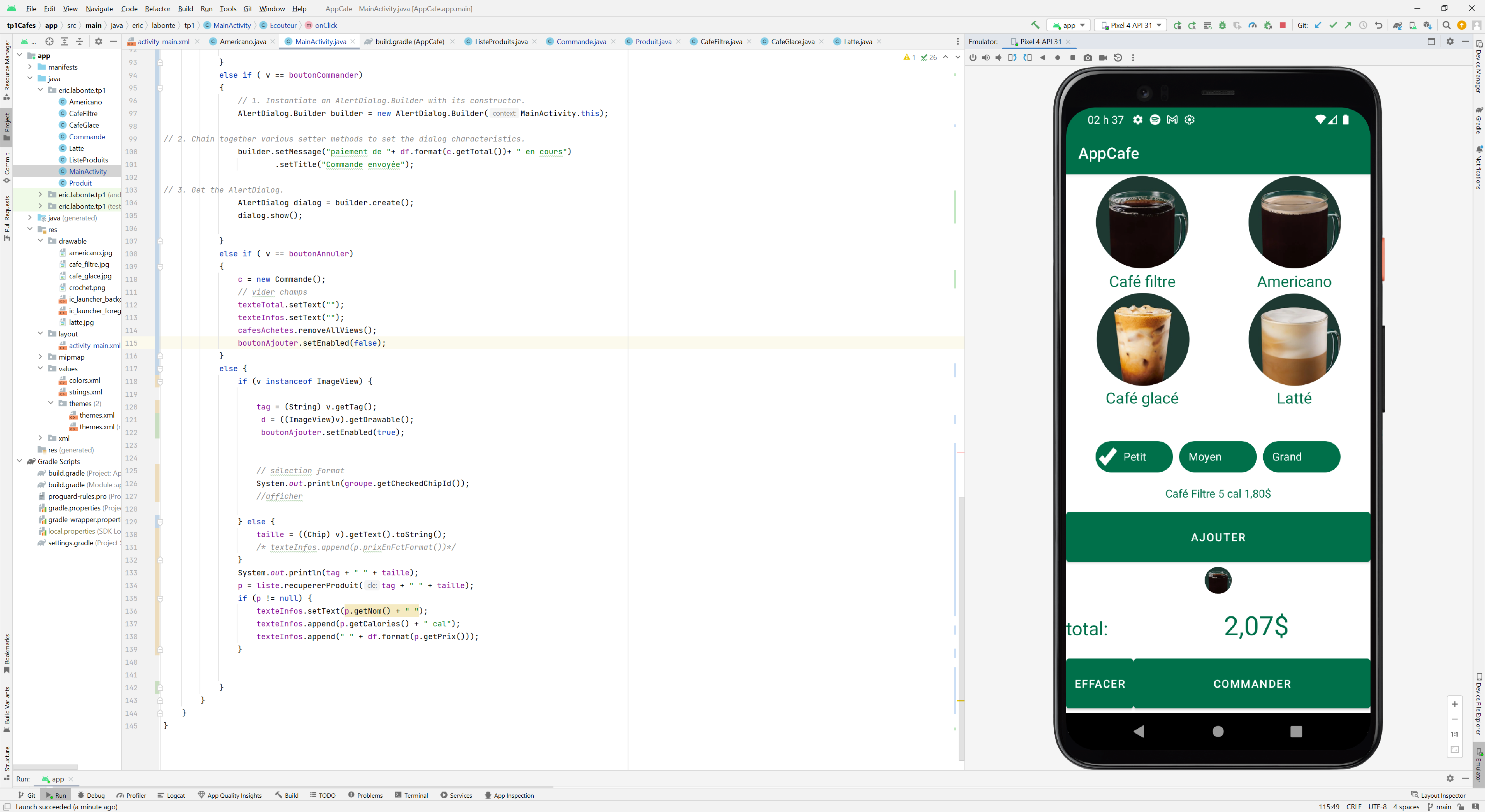
# TP1 – L'application des cafés Buckstar



Buckstar est une multinationale du café qui ouvrira sous peu ses premiers cafés à Montréal. L'ouverture des cafés sera de pair avec le démarrage d'une application Android permettant de commander son café à l'avance, le payer et aller le chercher de manière à ne pas avoir à faire la file ! Dans ce tp, vous aurez à développer une version de base de cette application.

## Le résultat visuel final :



*Le client ajoute à sa commande un petit Latté ; on voit le nouveau total ainsi qu'une image représentant le latté dans la commande*

*Le client a d'abord choisi un petit café filtre, on a affiché le nb de calories et le prix. Il clique sur Ajouter pour obtenir le total avec les taxes*

Petit-moyen-grand : 3 chip dans le chipgroup – choisir singleselection(event)

Linearlayout grav center : ajouter dynamiquement le drawable selectionné (customDrawing)

\*\*\* Pour vous donner une meilleure idée, regardez le video tp1.mp4 pour des utilisations des différents cas d'utilisation.

## Votre travail

Avant d'attaquer le dessin de la vue de l'application, on devra y associer un modèle objet représentant les produits vendus chez BuckStar de même que le contenu d'une commande d'un client

Voici les informations pertinentes à connaître sur les produits vendus :

MODELE

Constructeur initialise var prix, cal, en fonc du

Superclass (var commun – format,prix,nb cal..) & subclass (4)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Produits** | | | | |
| **nom** | **Formats possibles** | **Prix** | **Nb. de calories** |  |
| Café filtre | Petit | 1.80$ | 5 |  |
| Moyen | 5/3 du prix d'un petit café filtre | 7 |  |
| Grand | 2,2 fois le prix d'un petit café filtre | 2 fois le nombre de calories d'un petit café filtre |  |
| Americano | Petit | 2,40$ | 9 |  |
| Moyen | 5/3 du prix d'un petit americano | 2 calories de plus qu'un petit americano |  |
| Grand | 2,2 fois le prix d'un petit americano | 2 fois le nombre de calories d'un petit americano |  |
| Café glacé | Petit | 2,50$ | 10 |  |
| Moyen | 5/3 du prix d'un petit café glacé | 2 calories de plus qu'un petit café glacé |  |
| Grand | 2,2 fois le prix d'un petit café glacé | 2 fois le nombre de calories d'un petit café glacé |  |
| Latté | Petit | 4,00$ | 125 |  |
| Moyen | 5/3 du prix d'un petit café Latté | 5/3 du nb de calories d'un petit Latté |  |
| Grand | 2,5 fois le prix d'un petit Latté | 2 fois le nombre de calories d'un petit Latté |  |

1. Développez un modèle objet contenant une superclasse ainsi que des sous-classes représentant chacune un des produits ci-dessous. Ces sous-classes permettront de pouvoir créer des objets représentant ces boissons ainsi que pouvoir connaître leurs prix et leurs nombres de calories associés étant donné la grandeur du breuvage.
2. Ajoutez également un fichier / classe Commande représentant la commande en cours. Une commande aura 2 variables d'instance :

* Un Vector contenant les boissons que l'usager veut acheter
* Le total en argent de la commande

Codez un constructeur initialisant le Vector vide et le total = 0

Codez une méthode permettant d'ajouter un Produit acheté à la commande et d'augmenter le total en conséquence.

Codez des méthodes d'accès ( get ) si nécessaire

1. Intégrez la classe ListeProduits, présente sur LÉA, à votre projet. Cette classe comprend une table de hachage qui permettra de retrouver rapidement un produit de chez Buckstar à partir d'une clé qui sera :

Le nom du produit + le format du produit

Compléter la partie "élément" de la Hashtable en y créant des objets de votre modèle correspondant aux clés indiquées.

Codez le contenu de la méthode recupererProduit permettant de retourner l'élément correspondant à la clé passée en paramètre

1. Dessinez l'interface graphique de l'activité de manière à ce qu'elle soit similaire à celle de la page 1. Notez les éléments suivants :
2. Le conteneur principal ainsi que les sous-conteneurs doivent être des LinearLayouts.

1. Les items représentant des produits sont des ImageViews; ceux représentant les grandeurs sont des Chips ( contenu dans un conteneur ChipGroup ) . Faites en sorte qu'une seule sélection soit possible. Le format petit doit être sélectionné par défaut.

L'image crochet est présente sur LÉA. On peut modifier le style des objets Chip pour qu'il intègre le crochet lorsque sélectionné 🡪 voir explications en classe

1. Le bouton Ajouter ne doit pas être fonctionnel au départ ( propriété enabled à false ) Il le deviendra quand la personne a choisi un produit ( setEnabled(true))
2. Vous pouvez ajouter des ImageViews miniatures dynamiquement pour indiquer qu'un produit a été choisi par le client. Ceci peut être fait de la même façon qu'on ajoute une surface de dessin à un conteneur. Voir les détails à l'annexe 6.
3. Lisez sur les AlertDialogs afin d'afficher le message représentant l'envoi vers la procédure de paiement
4. Utilisez votre modèle objet pour obtenir le montant des achats de chaque produit ( sans les taxes ) ainsi que le total ( incluant les taxes )