

IFT1166 – TRAVAIL PRATIQUE #2 – 22 Février 2K

Une machine distributrice de boissons

Mohamed Lokbani

LA DATE LIMITE EST LE 15 Mars 2K Minuit, avec une pénalité de 2 points par jours de retard.

À faire en équipe de deux.

1. BUT

Ce TP a pour but de vous faire pratiquer les matières suivantes :

- * Programmation modulaire procédurale en C++
- * Les notions de C++ (classes, constructeurs/destructeurs, surcharge des opérateurs, fonctions amies etc.)

Conseils

- * Lisez l'énoncé de A à Z, puis en cas de problèmes contactez-nous.
- * N'attendez pas la semaine précédant la remise avant de commencer ... vous n'aurez pas le temps!

2. Description

Même si aujourd'hui la machine distributrice de boissons est présente un peu partout, son fonctionnement interne nous importe peu, même dans les cas où cette machine a "oublié" de retourner la monnaie, ou le produit sélectionné!! Dans ce travail pratique, vous allez concevoir une machine distributrice, du point de vue logiciel, tout en faisant bien sûr attention à correctement traiter les cas "d'oubli".

Pour éviter de compliquer ce travail, nous n'allons nous attarder qu'à un scénario simple du fonctionnement de cette machine:

Une personne entre des pièces de monnaie, puis sélectionne son produit. La machine fait les vérifications nécessaires relatives à la somme entrée ainsi que sur la disponibilité du produit. Si ces tests sont concluants, la machine retourne le produit à la personne, sinon il y aura un traitement adéquat en conséquence.

3. Travail à réaliser

Vous avez à écrire des programmes en C++ permettant de réaliser les opérations d'échanges entre les consommateurs et la machine distributrice de boissons. Pour ce faire, certains outils vous sont fournis et certaines contraintes vous sont imposées.

3.1 outils fournis

Nous mettons à votre disposition 6 fichiers, ces fichiers sont disponibles sur la page web du tp2 @:

<http://www.iro.umontreal.ca/~dift1166> sinon /u/dift1166/Sujet/H2K/tp2

Ces fichiers contiennent ce qui suit:

3.1.1 fichiers d'en-tête

`monnaie.h`: ce fichier contient la description de la classe `Monnaie`. Cette classe contient les outils nécessaires pour la création et la gestion d'une somme d'argent qu'une personne peut avoir.

`produit.h`: ce fichier contient la description de la classe `Produit`. Cette classe contient les outils nécessaires pour créer et gérer les produits disponibles dans un distributeur de boissons.

`db.h`: ce fichier contient la description de la classe `Distributeur`. Cette classe contient les outils nécessaires pour simuler un distributeur de boissons.

3.1.2 fichier programme

`prgtest.cpp`: ce fichier contient un programme de simulation du fonctionnement de la machine distributrice de boissons en présence de consommateurs. Il vous permettra de tester vos programmes.

3.1.3 fichiers de données

`simulation.dat`: un fichier de données d'entrée utilisées par l'exécutable de `prgtest.cpp`

`sortie.dat`: le résultat du programme exécutable de `prgtest.cpp`

Voir le contenu des fichiers `monnaie.h`, `produit.h`, `db.h` et `prgtest.cpp` pour la description complète de leur fonctionnement.

3.2 Contraintes

Vous ne devez tenir compte que des contraintes imposées dans les fichiers d'en-tête (`monnaie.h`, `produit.h` et `db.h`) et du programme de test (code source `prgtest.cpp`), mis à votre disposition.

Vous devez respecter les formats de sortie des résultats tel que décrits dans les fichiers d'en-tête (`monnaie.h`, `produit.h` et `db.h`) et le fichier `sortie.dat`

Par soucis de simplifier le problème, il n'est pas de votre rôle de prévoir s'il y a assez de monnaie disponible dans la machine distributrice de boissons ou assez de place pour stocker cette monnaie. Si la machine distribue du café, qu'il y a assez de café, que la dose à distribuer est bien correcte, qu'il y a bien un gobelet de disponible pour contenir ce café etc.

4. Programmes à remettre

Vous devez écrire la description complète des classes `Produit` `Monnaie` et `Distributeur` en tenant compte de l'organisation suivante:

fichier en-tête	fichier code source correspondant
<code>monnaie.h</code>	<code>monnaie.cpp</code>
<code>produit.h</code>	<code>produit.cpp</code>
<code>db.h</code>	<code>db.cpp</code>

Vous devez remettre les 3 fichiers: `monnaie.cpp`, `produit.cpp` et `db.cpp` voir le paragraphe 6 pour la procédure de remise.

5. Compilation

5.1. Compilation séparée

Après avoir inclus le fichier d'en-tête dans le fichier source correspondant, faites ce qui suit:

fichier en-tête	fichier source	commande de compilation
<code>monnaie.h</code>	<code>monnaie.cpp</code>	<code>g++ -Wall -c monnaie.cpp</code>
<code>produit.h</code>	<code>produit.cpp</code>	<code>g++ -Wall -c produit.cpp</code>
<code>db.h</code>	<code>db.cpp</code>	<code>g++ -Wall -c db.cpp</code>

chaque fichier source compilé, `xxx.cpp`, va générer s'il n'y a pas d'erreurs le fichier, `xxx.o`

5.2. compilation globale

Une fois les 3 fichiers compilés, liez-les ensemble avec la commande:

```
g++ -Wall -o Tdb db.cpp monnaie.cpp produit.cpp testprg.cpp
```

où Tdb est le nom du programme exécutable (vous pouvez choisir un autre nom).

6. Remise

Vous devez faire deux types de remise :

- Papier (à glisser sous la porte de mon bureau, Pavillon André-Aisenstadt, local 2249)
bien identifier sur le listing le cours (IFT1166) et vos noms.

- Électronique : dans le répertoire où sont vos programmes, tapez:

```
remise ift1166 tp2 monnaie.cpp produit.cpp db.cpp
```

7. Barème

Ce tp2 est noté sur 15 points. Les points seront répartis comme suit :

Exactitude : 7 points

un programme avec des avertissements "warnings" mais qui fait des choses non prévues dans la spécification 0/7,

un programme qui ne compile pas 0/7,

un programme avec des avertissements ("warnings") et qui donne les résultats escomptés 5/7.

Modularité : 6 points

la manière avec laquelle vous allez coder votre programme!

Style : 2 points

le code doit être lisible, bien indenté et commenté.

8. Des questions à propos de ce TP?

l'adresse email: dift1166@iro.umontreal.ca

pour faciliter le traitement de votre requête, inclure dans le sujet de votre email, au moins la chaîne: [IFT1166]

Les questions posées (ainsi que les réponses) durant la durée de ce tp seront archivées sur la page web du tp2. Vous pouvez les consulter on-line pour voir si votre question y figure déjà.

9. Mise à jour

22-02-2K diffusion

bon courage!