Cégep André-Laurendeau

420-235-AL (Hiver 2024)

Travail pratique 1

Classes, objets, constructeurs, copies, encapsulation, variables statiques, mutateurs et accesseurs, comparaison d'objets, surcharge de fonctions, gestion de versions et tests unitaires.

Échéance : 28 février à minuit

David Giasson et Michel Généreux

Objectif

Utiliser les techniques de programmation orientée objet vues dans le module 2 du cours pour concevoir les classes et méthodes répondants aux besoins d'une application logicielle.

Directives

- Ce travail peut être fait **seul ou en équipe de deux :**
  - Si le travail est fait en équipe, indiquez le nom des deux coéquipiers en commentaire en haut de chacun des fichiers de code source.
- Une fois terminé, le travail doit être remis sur Léa :
  - Compressez le répertoire qui contient le projet IntelliJ pour votre code
  - Déposez ce fichier .zip sur Léa (une seule remise par équipe)
  - Pour vérifier que la remise a bien fonctionné :
    - Téléchargez votre fichier .zip depuis Léa
    - Décompressez le fichier .zip
    - Ouvrez le projet dans IntelliJ et assurez-vous que tout fonctionne correctement
- Le travail doit être remis au plus tard à la date officielle indiquée sur Léa. Après cette date, 10% de pénalité sera enlevé par jour de retard.
- Toute forme de plagiat entrainera automatiquement la note zéro (0) ainsi qu'un rapport officiel à votre dossier. N'oubliez-pas d'indiquer vos références si vous utilisez des extraits de code trouvés sur internet ou provenant d'une autre personne.
- La qualité du français est également importante. Jusqu'à 10% de la note finale de votre travail pourrait être retirée pour cause d'un mauvais usage du français.

Critères d'évaluation

- Modélisation des classes demandées
- Fonctionnement correct du programme
- Qualité du code (noms des méthodes et variables, formatage, lisibilité, etc.)

Mise en contexte

Votre mandat est de coder un prototype des classes qui serviront à simuler le montage (une configuration) d'un ordinateur à partir de composants matériels, avec un œil sur les coûts que ça implique. Pour vous guider dans cette tâche complexe, le développeur sénior de votre équipe vous a fourni quelques indications sur les pages suivantes, ainsi que la classe *Main*, qui simulera le montage et le calcul des coûts.

Travail à réaliser

Classe 1 : Les composants (20%)

Un **composant** est définie par :

- Une **catégorie** (qui est constante et doit être formattée en majuscules).
- Une **marque** et un **nom**
- Un **prix** d'achat et un **rabais**, le cas échéant. Le prix et le rabais doivent être positifs. De plus, le rabais ne pourra excéder 100%.

En plus des accesseurs et mutateurs nécessaires, voici les méthodes de la classe:

- public Composant(String categorie, String marque, String nom, double prix) :
  - Le constructeur de la classe. Le rabais est mis à zéro à la création.
- public Composant copier() :
  - Méthode de copie de l'objet.
- public boolean estIdentique(Composant autre) :
  - Méthode pour vérifier si deux composants sont égaux sur la base de la catégorie, de la marque et du nom.
- public double getPrix() :
  - Cet accesseur devra retourner le prix **avec** le rabais.
- public String toString() :
  - Pour convertir un composant en String, comme ceci :
    - [catégorie] marque nom

Vous pouvez aussi ajouter n'importe quelle autre méthode ou attribut qui vous semble pertinent.

Critères de correction	**Points**
Attributs (types et noms appropriés)	3
Accesseurs et mutateurs	2
Constructeur	3
Méthode copier()	3
Méthode estIdentique()	3
Méthode getPrix()	3
Méthode *toString()* pour l'affichage	3

Classe 2 : Les configurations (60%)

Une **configuration** est définie par :

- Une **description** (qui est constante et doit être formattée en majuscules).
- Un **prix maximal** : la configuration ne pourra dépasser ce prix (quand on additionne le prix de chaque composant).
- Un tableau de **composants.**
- Le **nombre de composants** dans la configuration : ne peut dépasser **MAX_COMPOSANTS.**
- **MAX_COMPOSANTS** : une constante limitant le nombre de composants dans la configuration. Sa valeur est fixée à 20.

En plus des accesseurs et mutateurs nécessaires, voici les méthodes de la classe:

- public Configuration(String description, double prixMax, Composant[] composants)
    - Le constructeur principal de la classe
- public Configuration(Configuration originale) :
    - Le constructeur de copie permettant une copie profonde
- public double calculerTotal(double taxe) :
    - Cette méthode calcule le prix total de tous les composants, incluant la taxe
- public Composant rechercher(String categorie) :
    - Recherche (et retourne) un composant avec la catégorie **categorie**. Retourne **null** s'il n'y en a pas.
- public boolean ajouter(Composant composant) :
    - Ajoute un composant à la configuration, à condition qu'un composant de la même catégorie ne soit pas déjà inclus dans la configuration. Il ne faut pas non plus dépasser le nombre maximum de composants ou le prix maximal permis dans cette configuration. Si on ne peut faire l'ajout, on retourne **false**.
- public boolean retirer(Composant composant) :
    - Retire un composant à la configuration, à condition qu'il existe. En cas d'échecon retourne **false**.
- public boolean remplacer(Composant composant) :
    - Remplace un composant de la même catégorie. S'il n'y a pas de composants de la même catégorie, on retourne **false**.
- public String toString() :
    - Pour convertir une configuration en String, comme ceci :
        - description (prix maximal) :
            - 1 : composant (prix)
            - 2 : composant (prix)
            - etc.

Vous pouvez aussi ajouter n'importe quelle autre méthode ou attribut qui vous semble pertinent.

Critères de correction	Points
Attributs (types et noms appropriés)	9
Accesseurs et mutateurs	3
Constructeur principal	6
Constructeur de copie	6
Méthode calculerTotal()	6
Méthode rechercher()	6
Méthode ajouter()	6
Méthode retirer()	6
Méthode remplacer()	6
Méthode *toString()* pour l'affichage	6

Tests unitaires (8%)

Ajoutez une classe de tests JUnit5 nommée **TesterAjouter** pour tester la méthode **ajouter** de la classe **Configuration** et incluez les quatre cas suivants :

1. Ajouter un composant à la configuration pour que ça fonctionne.
2. Ajouter un composant à la configuration pour que ça ne fonctionne pas : il y a déjà un composant de la même catégorie dans la configuration.
3. Ajouter un composant à la configuration pour que ça ne fonctionne pas : on dépasse le prix maximal permis pour la configuration.
4. Ajouter un composant à la configuration pour que ça ne fonctionne pas : on dépasse le nombre de composants maximum permis pour la configuration.

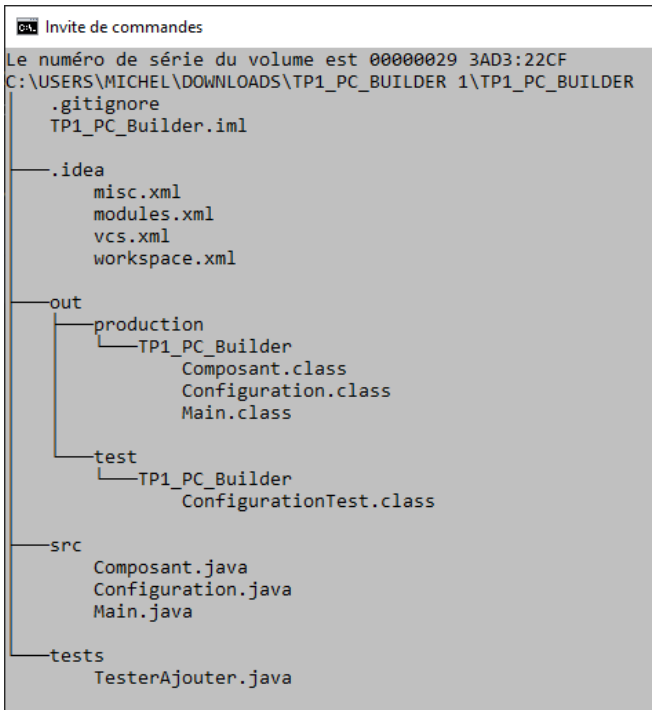
Git et GitHub (2%)

Faites régulièrement des **commit** durant votre travail. Une fois terminée, créez un dépôt public **420-TP1-H24** dans votre compte GitHub. Déposez vos fichiers sources .java dans ce dépôt et inscrivez l'adresse URL de ce dépôt dans l'entête de votre fichier Main.java (celui que vous aurez remis sur LÉA) : je pourrai ainsi jeter un coup d'œil à vos différents commit.

Remise (5%)

Une fois terminé, remettez votre travail sur Léa dans un fichier zip contenant :

- Dossier TP1_VotreNom (racine)
  - Fichier TP1.iml (ou portant nom similaire) pour votre **projet IntelliJ**
  - Dossier « **src** » contenant vos classes *Configuration* et *Composant* ainsi que la classe Main qui vous a été fournie dans le fichier de départ. Le fichier *Main* ne devrait pas avoir été modifié (sauf pour y ajouter votre ou vos noms en haut et votre URL GitHub) car lors de la correction il sera remplacé par notre version de départ.
  - Les autres dossiers ne sont pas nécessaires mais peuvent être remis quand même.
  - Voici l'allure de votre zip remis une fois décompressé :



Annexe : Exemple du résultat attendu

```
=== Test #1 : Affichage des configurations initiales ===

Build vide :

  Total: 0 composants pour 0,00$ (taxes incluses)

Build Intel Gen13 (max 1250,00$) :
   1: [CPU] Intel Core i5-13600K (330,00$)
   2: [CARTE MÈRE] Asus ROG Strix B760 (200,00$)
   3: [RAM] GSkill Trident-Z DDR5 16GB (90,00$)
   4: [GPU] Asus RTX 4060 (460,00$)

  Total: 4 composants pour 1242,00$ (taxes incluses)
```

```
Build AMD Gen5 (max 1000,00$) :
   1: [CPU] AMD Ryzen 5 5700X (260,00$)
   2: [CARTE MÈRE] MSI B550 Gaming Wifi (150,00$)

  Total: 2 composants pour 471,50$ (taxes incluses)


=== Test #2 : Recherche et manipulation de composants ===

La composante: [CPU] Intel Core i5-13600K
Sa catégorie: CPU
Est-ce core13600k?: true
La composante: [CARTE MÈRE] MSI B550 Gaming Wifi
Son prix: 150.0
Son rabais: 0.0
La composante: [RAM] GSkill Trident-Z DDR5 16GB
Sa catégorie: RAM
La prix de la composante: 75.0
Son rabais: 0.25
Est-ce tridentzDDR5?: true
La composante: null

=== Test #3 : Ajouts de composants ===

[RAM] GSkill Trident-Z DDR4 32GB ajouté à la configuration (total=545,00$)
[SSD] Western Digital SN850X 1TB ajouté à la configuration (total=645,00$)
# Rabais de 20% sur [GPU] Gigabyte RTX 4060 (nouveau prix: 320,00$)
[GPU] Gigabyte RTX 4060 ajouté à la configuration (total=965,00$)

Build AMD Gen5 (max 1000,00$) :
   1: [CPU] AMD Ryzen 5 5700X (260,00$)
   2: [CARTE MÈRE] MSI B550 Gaming Wifi (150,00$)
   3: [RAM] GSkill Trident-Z DDR4 32GB (135,00$)
   4: [SSD] Western Digital SN850X 1TB (100,00$)
   5: [GPU] Gigabyte RTX 4060 (320,00$)

  Total: 5 composants pour 1109,75$ (taxes incluses)

Il y a déjà un composant de cette catégorie: [CPU] AMD Ryzen 5 5700X
Il y a déjà un composant de cette catégorie: [GPU] Gigabyte RTX 4060
L'ajout de ce composant ferait dépasser le prix maximum: [SSD] Samsung 980 Pro 2TB

=== Test #4 : Retrait de composants ===

[RAM] GSkill Trident-Z DDR4 32GB retiré de la configuration (total=830,00$)
[GPU] Gigabyte RTX 4060 retiré de la configuration (total=510,00$)
[CPU] AMD Ryzen 5 5700X retiré de la configuration (total=250,00$)

Build AMD Gen5 (max 1000,00$) :
   1: [CARTE MÈRE] MSI B550 Gaming Wifi (150,00$)
   2: [SSD] Western Digital SN850X 1TB (100,00$)
```

```
  Total: 2 composants pour 287,50$ (taxes incluses)

Composant introuvable: [RAM] GSkill Trident-Z DDR4 32GB
Composant introuvable: [GPU] Gigabyte RTX 4060
Composant introuvable: [CPU] AMD Ryzen 5 5700X

=== Test #5 : Remplacement de composants ===

[SSD] Western Digital SN850X 1TB retiré de la configuration (total=150,00$)
[SSD] Samsung 980 Pro 2TB ajouté à la configuration (total=400,00$)
Build AMD Gen5 (max 1000,00$) :
   1: [CARTE MÈRE] MSI B550 Gaming Wifi (150,00$)
   2: [SSD] Samsung 980 Pro 2TB (250,00$)

  Total: 2 composants pour 460,00$ (taxes incluses)


=== Test #6 : Copie de configurations ===

Build AMD Gen5 (max 1000,00$) :
   1: [CARTE MÈRE] MSI B550 Gaming Wifi (99,00$)
   2: [SSD] Samsung 980 Pro 2TB (250,00$)

Build AMD Gen5 (copie) (max 1000,00$) :
   1: [CARTE MÈRE] MSI B550 Gaming Wifi (150,00$)
   2: [SSD] Samsung 980 Pro 2TB (250,00$)


=== Test #7 : Validation finale ===

Build Intel Gen13 (max 1250,00$) :
   1: [CPU] Intel Core i5-13600K (330,00$)
   2: [CARTE MÈRE] Asus ROG Strix B760 (200,00$)
   3: [RAM] GSkill Trident-Z DDR5 16GB (90,00$)
   4: [GPU] Asus RTX 4060 (460,00$)

  Total: 4 composants pour 1242,00$ (taxes incluses)

Build AMD Gen5 (max 1000,00$) :
   1: [CARTE MÈRE] MSI B550 Gaming Wifi (99,00$)
   2: [SSD] Samsung 980 Pro 2TB (250,00$)

  Total: 2 composants pour 401,35$ (taxes incluses)

Build AMD Gen5 (copie) (max 1000,00$) :
   1: [CARTE MÈRE] MSI B550 Gaming Wifi (150,00$)
   2: [SSD] Samsung 980 Pro 2TB (250,00$)

  Total: 2 composants pour 460,00$ (taxes incluses)
```