

Le but de ce TP est de programmer, en C#, une petite application console simple.

TP n°2

Codage application simple

Alexandre GUIDET

Compétence 1	Compétence 2
<input checked="" type="checkbox"/> Implémenter des conceptions simples <input type="checkbox"/> Élaborer des conceptions simples <input checked="" type="checkbox"/> Faire des essais et évaluer leurs résultats	<input type="checkbox"/> Analyser un problème avec méthode
<i>En respectant les besoins décrits par le client</i> <i>En appliquant les principes algorithmiques</i> <i>En veillant à la qualité du code et à sa documentation</i> <i>En choisissant les ressources techniques appropriées</i>	<i>En formalisant et modélisant des situations complexes</i> <i>En recensant les algorithmes et les structures de données usuels</i> <i>En s'appuyant sur des schémas de raisonnement</i> <i>En justifiant les choix et validant les résultats</i>

Lors de ce TP, nous allons programmer le fonctionnement d'une simulation de facturation d'impressions/photocopies.

Le centre de photocopie facture :

- 0,25 EUR / page pour les 10 premières photocopies (dans la même impression)
- 0,20 EUR / page pour les vingt suivantes
- 0,10 EUR / page au-delà.

Le code produit sera renvoyé à votre enseignant (voir A renvoyer).

Avec Visual Studio 2022, créez un nouveau projet de type « Application console », en C#.

1 CRÉATION DE LA CLASSE

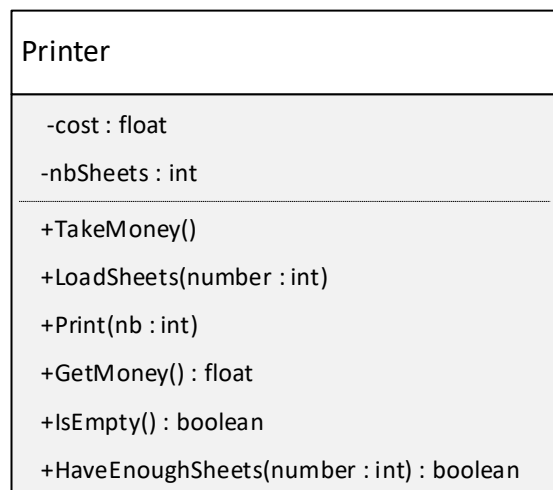
Nous souhaitons créer une classe `Printer` qui représentera la photocopieuse. Le diagramme UML ci-dessous représente la classe : respectez les noms indiqués.

Cette classe contiendra les **attributs** suivants :

- Nombre de feuilles disponibles
- Montant contenu dans la photocopieuse

Le **comportement** attendu de la classe est le suivant :

- **Relève** de la caisse de la photocopieuse (`TakeMoney`) → mise du montant à 0.
- **Charge de la photocopieuse** (`LoadSheet`), action permettant d'ajouter le nombre de feuilles passé en paramètre dans la photocopieuse
- **Imprimer** (`Print`), action qui photocopie le nombre de pages demandées (si possible : s'il y a moins de feuilles, elle ne photocopie que le nombre de feuille) et met à jour le montant et le nombre de feuilles
- **Accesseurs** permettant de savoir si la photocopieuse est vide (`IsEmpty`), si elle contient suffisamment de feuilles (`HaveEnoughSheets`), et quel est le coût total (`GetMoney`).



A la création de l'objet, il y a 0 feuilles et un coût de 0.

Créez cette classe en indiquant, pour l'instant, les attributs et le constructeur (qui ne possède aucun paramètre).

2 ACCESSEURS

Codez les accesseurs (opérations *non mutables* renseignant sur l'**état** de l'objet), sachant que :

- IsEmpty renvoie **vrai** si la photocopieuse ne contient **aucune** feuille
- HaveEnoughSheets renvoie **vrai** si la photocopieuse contient suffisamment de feuilles pour fournir la valeur demandée.
- GetMoney renvoie la valeur actuelle du coût.

Testez dans le programme principal :

- Créez un objet Printer
- Vérifiez que IsEmpty renvoie vrai
- Vérifiez que HaveEnoughSheets renvoie **faux** quelle que soit la valeur demandée (sauf si cette dernière est nulle)
- Vérifiez que GetMoney renvoie **0**.

Le programme sera modifié par la suite, et nous verrons plus tard comment tester une fonction sans modifier le programme.

3 MODIFICATEURS

Codez les différents modificateurs (opérations modifiant l'état de l'objet).

3.1 TAKEMONEY

Cette opération remet à zéro le coût. Codez-là, mais vous ne pourrez pas la tester pour l'instant.

3.2 LOADSHEETS

Cette opération ajoute des feuilles. Codez-là et testez-là dans un petit programme simple (gardez votre test précédent, mais mettez-le en commentaire) :

- Créez un objet Printer
- Chargez 100 feuilles
- Vérifiez que HaveEnoughSheets(100) renvoie **vrai** mais que HaveEnoughSheets(101) renvoie **faux**
- Chargez 10 feuilles
- Vérifiez que HaveEnoughSheets(101) renvoie **vrai** à présent, mais HaveEnoughSheets(111) renvoie **faux**.

3.3 PRINT

Cette opération simule une impression. Elle doit donc, suivant le nombre de feuilles demandées, modifier le coût et le nombre de feuilles restantes.

Attention : s'il est demandé 100 feuilles et qu'il en reste 80, seules 80 feuilles sont imprimées et facturées, les autres sont abandonnées.

- ① *Conseil : utilisez une variable locale pour stocker le nombre de feuilles au coût 0.25, une autre pour le nombre de feuilles au coût 0.2, une autre pour le nombre de feuilles au coût 0.1.*
- ① *Astuce : la méthode `Min` de la classe `Math` (`Math.Min`) permet d'avoir la plus petite valeur entre deux valeurs, sans avoir besoin de passer par une alternative...*

Pour tester, modifiez le programme principal (gardez votre test précédent, mais mettez-le en commentaires) pour avoir les actions suivantes :

- Créez un objet `Printer`
- Chargez l'objet avec 100 feuilles
- Imprimez 10 pages
- Vérifiez que le coût actuel est de 2,5€
- Imprimez ensuite 15 pages
- Vérifiez que le coût actuel est de 6 €
- Imprimez ensuite 40 pages
- Vérifiez que le coût actuel est de 13.5 €
- Relevez l'argent (`TakeMoney`)
- Vérifiez que le coût actuel est de 0 €.
- Imprimez ensuite 40 pages
- Vérifiez que `IsEmpty` est bien égal à **true**
- Vérifiez que le coût actuel est de 7€

4 PROGRAMME PRINCIPAL

Nous souhaitons le programme suivant : la faculté possède deux photocopieuses : une en salle des professeurs et une à destination des étudiants à la bibliothèque. Au début du programme, les deux photocopieuses sont chargées avec 1000 pages.

Simuler dans votre programme la séquence suivante :

- Un.e. enseignant.e. demande 200 photocopies dans la salle des professeurs.
- Un.e étudiant.e. retire 12 photocopies à la bibliothèque.
- Un.e. enseignant.e. demande 500 photocopies dans la salle des professeurs.
- Un.e étudiant.e. retire 30 photocopies à la bibliothèque.
- Un.e. enseignant.e. demande 400 photocopies dans la salle des professeurs.
- Un.e étudiant.e. retire 3 photocopies à la bibliothèque.

Affichez à l'écran le coût pour chaque photocopieuse à la fin de ces opérations.

5 A RENVoyer

Envoyez par courriel à votre enseignant :

- Le fichier .CS contenant la classe Printer
- Le fichier .CS contenant le programme principal (avec les tests et le programme final).
- Indiquez dans le corps du mail ce qu'affiche votre programme principal (coût de chaque photocopieuse)