Tp #2 Application Caméra

OBJECTIFs

* Construire des classes utilisant d’autres classes.
* Implémenter des interfaces.
* Lever des exceptions et utiliser des types par énumération.
* Vérifier le fonctionnement des classes avec des tests unitaires fournis.

Description DE l’APPLICATION

L’application simule le fonctionnement d’une caméra numérique. La caméra possède ses propres paramètres. Elle peut contenir différentes cartes mémoire pour stocker des photos. On peut modifier la qualité, la sensibilité et la dimension des photos. Un flash peut être activé ou non pour une photo.



Carte mémoire qui est présentement insérée dans la caméra.

Pour résoudre ce problème, vous devrez définir les classes suivantes :

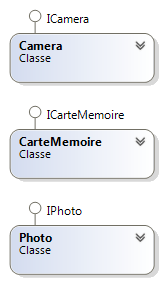
Le fichier **parametres.txt** contient les caracté­ris­ti­ques de la caméra :

0 🡨 la dimension est maximale

0 🡨 la qualite est excellente

1 🡨 la sensibilité est ISO\_100

1 🡨 le flash est désactivé



Le fichier **Sandisk 32Mo.txt** contient les données de chaque photo stockées sur la carte mémoire dont le nom est ***Sandisk 32Mo***:

**4000,3000,0,1,4,2010-02-14 17:08:23**

**3840,2160,0,1,50,2010-02-14 17:09:12**

. . .

largeur en pixel, hauteur en pixel, flash, sensibilité, facteur de compression, date heure

**Étape #0**

Prenez connaissance des types par énumération décrits dans le fichier **Enum.cs** :

* La **sensibilité** d’une photo 🡪 {ISO\_64, ISO\_100, ISO\_200, ISO\_400, ISO\_800}
* L’activation du **flash** 🡪 { Active, Desactive }
* La **qualité** d’une photo 🡪 { Excellente, Bonne, Moyenne, Faible }
* La **dimension** d’une photo 🡪 { Maximale, Large, Grande, Moyenne, Petite }

**Étape #1 – Créer votre repo. Sur Github et rajouter le projet de départ**

1. Si ce n’est pas fait, vous devez créer un compte GitHub avec votre compte du cégep (matricule@cegepmontpetit.ca ). Regardez la capsule vidéo sur Moodle et accéder à cette page pour le créer et avoir le Student Service Pack : <https://education.github.com/pack>
2. Connectez-vous à GitHub avec votre compte du cégep (matricule@cegepmontpetit.ca)
3. Créez un nouveau dépôt GitHub privé et le nommez selon ce format en mettant votre propre matricule au lieu de XXXXXXX

H23-CN4-TP02-XXXXXXX

**N.B.** Exceptionnellement, dans notre cours on va utiliser ce format. Sinon, pour les autres cours, vous devez faire référence à la page Git du département et les consignes de votre professeur ! <https://info.cegepmontpetit.ca/git>

1. Ajoutez votre professeur comme collaborateur à votre projet. (Sur votre repository, Onglet settings, Manage Access, Add a collaborator).

[jamil.gammoudi@cegepmontpetit.ca](mailto:jamil.gammoudi@cegepmontpetit.ca)

1. Avec GitKraken, cloner votre repo., rajouter le projet du départ et pousser !

**Étape #1**

* Définir la classe **Photo:IPhoto** telle que décrite ci-après.

|  |  |  |
| --- | --- | --- |
| **Photo:IPhoto** | | |
| Une photo est caractérisée par sa taille en pixels (largeur, hauteur), le flash activé ou non, la sensibilité, le facteur de compression ainsi que la date et l’heure où elle a été prise. | | |
|  | | Copiez les déclarations suivantes dans la classe Photo  public const int TAILLE\_PIXEL = 3;  public const int LARGEUR\_MIN = 640;  public const int LARGEUR\_MAX = 8192;  public const int HAUTEUR\_MIN = 480;  public const int HAUTEUR\_MAX = 8192;  public const int COMPRESSION\_MIN = 1;  public const int COMPRESSION\_MAX = 500; |
| **Propriétés/Méthodes** | | **Description** |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789784900/guicontroles/guitimer/Propri%C3%A9t%C3%A9.png | **Date** | Obtient la date et l’heure où la photo a été prise. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789784900/guicontroles/guitimer/Propri%C3%A9t%C3%A9.png | **FacteurCompression** | Obtient le facteur de compression de la photo. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789784900/guicontroles/guitimer/Propri%C3%A9t%C3%A9.png | **Flash** | Obtient si le flash est activé ou désactivé sur la photo. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789784900/guicontroles/guitimer/Propri%C3%A9t%C3%A9.png | **Sensibilite** | Obtient la sensibilité utilisée pour la photo. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789784900/guicontroles/guitimer/Propri%C3%A9t%C3%A9.png | **TailleEnPixels** | Obtient la taille en pixels de la photo.   Le type de cette propriété est Size. Elle aura une hauteur et une largeur de disponible.  ( m\_tailleEnPixels.Height donne la hauteur de la photo,  m\_tailleEnPixels.Width donne la largeur de la photo ). |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789784900/guicontroles/guitimer/Propri%C3%A9t%C3%A9.png | **TailleEnOctets** | Calcule la taille en octets de la photo à l’aide de la formule suivante :  *largeur en pixel* ***X*** *hauteur en pixel*  ***X*** *taille d’un pixel*  *facteur de compression*  La taille d’un pixel est de 3 octets. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789050330/instanciation_objets/M%C3%A9thode.png | **Photo** | Photo(Size pTailleEnPixels, int pFacteurCompression,  enuSensibilite pSensibilite, enuFlash pFlash)  Initialise une instance avec les données passées en paramètres.  Ce constructeur doit lever l’exception ArgumentOutOfRangeException lorsque les arguments possèdent des valeurs non acceptées par cette classe. Voyez les constantes qui vous donnent les limites acceptées. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789050330/instanciation_objets/M%C3%A9thode.png | **Photo** | Photo(string pInfosPhoto)  Initialise une instance à partir d’une chaîne dont les données sont séparées par une virgule.  Le format des données est le suivant : largeur et hauteur en pixel, flash, sensibilité, facteur de compression, date hre  Exemple: 4000,3000,1,1,4,2010-02-14 21:00🡨"yyyy-MM-dd HH:mm:ss" |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789050330/instanciation_objets/M%C3%A9thode.png | **ToString** | Retourne la représentation de cette instance sous forme de chaîne dont les données sont séparées par des virgules .  Exemple: 4000,3000,1,1,4,2010-02-14 21:00🡨"yyyy-MM-dd HH:mm:ss" |

**Tester votre classe en activant les tests : A1 à A5**

**Étape #2**

* Définir la classe **CarteMemoire:ICarteMemoire** telle que décrite ci-après.

|  |  |  |
| --- | --- | --- |
| **CarteMemoire:ICarteMemoire** | | |
| Une carte mémoire est définie par les caractéristiques suivantes :   * son **nom**. Par exemples : "Sandisk 32Mo", "UltraII 1Go" ou "SDHC 8Go" * sa **taille** en octets * **l’espace** **utilisé** en octets * la **collection** **de photos** stockées sur la carte. | | |
|  | | Copier les déclarations suivantes dans la classe CarteMemoire  public const int TAILLE\_MIN = 8; // 8 Mo  public const int TAILLE\_MAX = 8192; // 8192 Mo  public const string ERR\_MSG\_ESPACE\_INSUFFISANT = "Espace insuffisant"; |
| **Propriétés/Méthodes** | | **Description** |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789784900/guicontroles/guitimer/Propri%C3%A9t%C3%A9.png | **Nom** | Obtient le nom de la carte mémoire. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789784900/guicontroles/guitimer/Propri%C3%A9t%C3%A9.png | **TailleEnOctets** | Obtient la taille en octets de la carte mémoire. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789784900/guicontroles/guitimer/Propri%C3%A9t%C3%A9.png | **EspaceUtilisé** | Obtient l’espace utilisé en **octets**. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789784900/guicontroles/guitimer/Propri%C3%A9t%C3%A9.png | **NbPhotos** | Obtient le nombre de photos actuellement stockées dans la collection. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789784900/guicontroles/guitimer/Propri%C3%A9t%C3%A9.png | **EspaceDisponible** | Calcule l’espace en octets disponible sur la carte mémoire. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789050330/instanciation_objets/M%C3%A9thode.png | **CarteMemoire** | CarteMemoire(string pNom, int pTailleEnMegaOctets)  Pour construire une carte, on précisera le nom de la carte et sa taille en **MegaOctets**.  Les valeurs limites pour la taille en MegaOcters sont de 8 Mo et 8192Mo.  Une exception du type ArgumentOutOfRangeException() sera levée si ces limites ne sont pas respectées.  Il faudra transformer la taille en méga octets pour une taille en octets. Vous multiplierez simplement la taille en méga octets par 1024L \* 1024. (Voyez sur Moodle la raison pour laquelle faut mettre un L).  Il faudra également instancier la collection de IPhoto de cette carte mémoire.  Et au départ, aucun espace n’est utilisé sur la carte. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789050330/instanciation_objets/M%C3%A9thode.png | **ChargerPhotos** | Permet de stocker dans la collection de photos, les données qui sont dans le fichier dont le nom est formé à l’aide du nom de la carte et du suffixe .txt  Ajoutez le namespace nécessaire pour pouvoir lire et écrire dans des fichiers. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789050330/instanciation_objets/M%C3%A9thode.png | **EnregistrerPhotos** | Permet de sauvegarder les photos de la collection vers un fichier dont le nom est formé à l’aide du nom de la carte et du suffixe .txt |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789050330/instanciation_objets/M%C3%A9thode.png | **PeutAjouter** | Indique si oui ou non, l’espace disponible est suffisant pour éventuellement ajouter la photo sur la carte. Il faut aussi vérifier que la photo qu’on veut ajouter n’est pas null. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789050330/instanciation_objets/M%C3%A9thode.png | **Ajouter** | Si c’est possible, ajoutez la photo à la collection de photos sinon lancez une exception de type InvalidOperationException.  Une fois la photo ajoutée à la collection, n’oubliez pas d’ajuster l’espace utilisé sur la carte. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789050330/instanciation_objets/M%C3%A9thode.png | **SupprimerAt** | Supprime de la collection, la photo dont la position est indiquée en paramètre.  On doit s’assurer que la position est valide. . Si ce n’est pas possible lancez un ArgumentOutOfRangeException.  Une fois la photo ajoutée à la collection, n’oubliez pas d’ajuster l’espace utilisé sur la carte. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789050330/instanciation_objets/M%C3%A9thode.png | **PhotoAt** | Permet d’obtenir l’objet photo dont la position est indiquée en paramètre. Si ce n’est pas possible lancez un ArgumentOutOfRangeException. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789050330/instanciation_objets/M%C3%A9thode.png | **Vider** | Vide complètement la collection de photos et ajuste l’espace utilisé sur la carte mémoire |

**Tester votre classe en activant les tests : B1 à B8**

**Étape #3**

* Définir la classe **Camera:ICamera** telle que décrite ci-après.

|  |  |  |
| --- | --- | --- |
| **Camera:ICamera** | | |
| La caméra possède des paramètres avec lesquels seront prises les photos. Ces paramètres pourront être lus ou sauvegardés dans le fichier dont le nom est « **parametres.txt** ». Les caractéristiques de la caméra sont les suivantes : la dimension , la qualite , la sensibilité et le flash est activé ou désactivé. | | |
|  | | Copier les déclarations suivantes dans la classe Camera  public const string ERR\_MSG\_CARTE\_MANQUANTE = "Carte manquante";  public const string ERR\_MSG\_CARTE\_PRÉSENTE = "Carte déjà présente";  public const string ERR\_MSG\_CARTE\_VIDE = "Carte vide";  privateconststringNOM\_FICHIER\_PARAMETRES = "parametres.txt"; |
| **Propriétés/Méthodes** | | **Description** |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789784900/guicontroles/guitimer/Propri%C3%A9t%C3%A9.png | **Dimension** | Obtient et définit le paramètre dimension de la caméra. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789784900/guicontroles/guitimer/Propri%C3%A9t%C3%A9.png | **Qualite** | Obtient et définit le paramètre qualité de la caméra. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789784900/guicontroles/guitimer/Propri%C3%A9t%C3%A9.png | **Sensibilite** | Obtient et définit le paramètre sensibilité de la caméra. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789784900/guicontroles/guitimer/Propri%C3%A9t%C3%A9.png | **Flash** | Obtient et définit le paramètre flash de la caméra. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789784900/guicontroles/guitimer/Propri%C3%A9t%C3%A9.png | **CarteMemoireCourante** | Obtient la carte mémoire qui est actuellement dans la caméra.   Au départ, comme il n’y en a pas, vous pouvez lui donner la valeur null. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789784900/guicontroles/guitimer/Propri%C3%A9t%C3%A9.png | **PosPhotoCourante** | Indique l’indice de la photo actuellement affichée sur la caméra. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789784900/guicontroles/guitimer/Propri%C3%A9t%C3%A9.png | **PhotoCourante** | Retournera l’objet photo contenu dans la collection de la carte mémoire courante à la position indiquée par la position courante.   Si ce n’est pas possible parce qu’il n’y a pas de photo dans la carte mémoire courante ou qu’il n’y a pas de carte mémoire courante dans la caméra, retourne null. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789784900/guicontroles/guitimer/Propri%C3%A9t%C3%A9.png | **FacteurCompression** | Le facteur de compression est déterminé à partir de la qualité sélectionnée de la caméra.  Il servivra à créer une photo directement à partir de la caméra.   Vous devrez créer un tableau tel que décrit ci-après :   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Qualité🡪 | Excellente (0) | Bonne (1) | Moyenne (2) | Faible  (3) | | facteur | 4 | 12 | 20 | 50 |   La qualité de la photo sera l’index nous permettant d’accéder au bon facteur de compression dans le tableau que vous aurez créé. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789784900/guicontroles/guitimer/Propri%C3%A9t%C3%A9.png | **TailleEnPixelsSelonDimension** | La taille en pixels selon la dimension est obtenue grâce au tableau suivant :   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Dimension🡪 | Maximale (0) | Large (1) | Grande (2) | Moyenne  (3) | Petite  (4) | | largeur | 4000 | 3840 | 3000 | 1600 | 640 | | hauteur | 3000 | 2160 | 2000 | 1200 | 480 |   La largeur et la hauteur constitue les informations nécessaires pour créer un objet Size.  Le tableau que vous créerez ici est un tableau d’objets Size. Vous créerez chacun des éléments de ce tableau en instanciant un nouvel objet Size avec la largeur et la hauteur donnée.  Ainsi le premier objet du tableau de Size sera : new Size(4000,3000)  Cette taille en pixels (Size) servira à créer une photo directement à partir de la caméra.  La dimension fait référence à la taille en pixels de la photo et sera l’index nous permettant d’accéder à la bonne taille de photo dans le tableau que vous aurez créé. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789784900/guicontroles/guitimer/Propri%C3%A9t%C3%A9.png | **TailleEnOctetsEstimee** | Calculez la taille de la photo selon la formule suivante :  *largeurEnPixel* ***X*** *hauteurEnPixel* ***X*** *taille d’un pixel*  *facteur de compression de la caméra*  Notez qu’il s’agit des largeur et hauteur **en pixel selon la dimension.** |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789050330/instanciation_objets/M%C3%A9thode.png | **Camera** | Camera()  Initialement la caméra ne contient pas de carte mémoire.  La position de la photo courante est à -1, la dimension est Maximale, qualité est Excellente, sensibilité est ISO\_64, et le flash est activé. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789050330/instanciation_objets/M%C3%A9thode.png | **ChargerParametres** | Permet de lire les valeurs des paramètres dans le fichier et d'initialiser les 4 variables membres suivantes : dimension, qualité, sensibilité et flash. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789050330/instanciation_objets/M%C3%A9thode.png | **EnregistrerParametres** | Permet de sauvegarder les 4 paramètres de la caméra dans le fichier. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789050330/instanciation_objets/M%C3%A9thode.png | **InsererCarteMemoire** | Définit la référence de la carte mémoire courante. Comme cette méthode simule l’insertion d’une carte (pCarteMemoire) dans la caméra, il y a des problèmes à prévenir :  Si la carte mémoire qu’on essaie d’insérer est null, lancer ArgumentNullException avec *Carte manquante*.    S’il y a déjà une carte courante dans la caméra, lancer InvalidOperationException avec message *Carte déjà presente***.**  Une fois la carte insérée, assurez vous que la position de la photo courante soit initialisée à 0. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789050330/instanciation_objets/M%C3%A9thode.png | **EjecterCarteMemoire** | Comme cette méthode simule l’éjection de la carte mémoire, il y a un problème à prévenir :  S’il n’y a pas de carte courante présentement dans la caméra, lancez InvalidOperationException avec message *Carte manquante*  Met la variable membre carte mémoire à null et la position de la photo courante à -1. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789050330/instanciation_objets/M%C3%A9thode.png | **ViderLaCarte** | Permet de vider la carte mémoire actuellement dans la caméra.  Comme cette méthode simule vider la carte mémoire, il y a un problème à prévenir :  S’il n’y a pas de carte courante présentement dans la caméra, lancez InvalidOperationException avec message *Carte manquante*  Vous utiliserez la méthode Vider() de la classe CarteMemoire pour la vider. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789050330/instanciation_objets/M%C3%A9thode.png | **PeutPrendreUnePhoto** | Indique s’il est possible de prendre une photo actuellement avec les paramètres courant de la caméra.  Donc on peut prendre une photo s’il y a une carte mémoire présentement dans la caméra et s’il y a suffisament de place disponible sur la carte mémoire courante pour la taille estismée en octets de la photo qu’on essaie de prendre. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789050330/instanciation_objets/M%C3%A9thode.png | **PrendrePhoto** | Crée une photo avec les paramètres actuels de la caméra.  Il faut s’assurer qu’il y a bien une carte mémoire dans la caméra, sinon l’exception est InvalidOperationException *CarteManquante*.  De plus, on s’assure que l’on peut ajouter la photo, sinon c’est l’exception InvalidOperationException *EspaceInsuffisant*.  Si on peut ajouter la photo à la carte mémoire courante, on le fait. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789050330/instanciation_objets/M%C3%A9thode.png | **SupprimerPhotoCourante** | On s’assure qu’il y a bien une carte dans la caméra, sinon l’exception est InvalidOperationException *CarteManquante*.  Vérifier si carte est vide et lancer InvalidOperationException *CarteVide*  Si on peut supprimer la photo de la carte mémoire courante, on le fait. |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789050330/instanciation_objets/M%C3%A9thode.png | **DimensionSuivante**  **QualiteSuivante** **SensibiliteSuivante** | Permet de passer à la prochaine valeur dans leurs émum respectifs.  Lorsqu’on arrive au bout, on revient à la 1ère . |
| https://sites.google.com/site/notionscsharpcem/_/rsrc/1370789050330/instanciation_objets/M%C3%A9thode.png | **PhotoPrecedente**  **PhotoSuivante** | Avance à la photo suivante ou recule à la photo précédente. Se fait de façon circulaire…  Dans les 3 cas, il faut vérifier qu’il y a une carte mémoire courante et qu’il y a des photos sur la carte. Autrement lancer les exceptions appropriées. |

**Tester votre classe en activant les tests : C1 à C8**

ÉVALUATION

* Fonctionnement des tests pour les 3 classes à créer
* La qualité de code peut diminuer la note du programme de correction jusqu'à 15%.
* Soigner les commentaires pour les classes créées🡪 jusqu’au **15%** pourrait être enlevé.
* Noter que le fonctionnement du formulaire est facultatif, il compte pour 0%