

420-4B6-LI Applications Web I Cégep Limoilou Département d'Informatique Hiver 2020	Tp1(10 %) Cocktail C#
---	--

Capacité 2 : Programmer la logique applicative côté serveur

OBJECTIFS

- ✦ Utiliser efficacement l'environnement Visual Studio.
- ✦ Connaître les bases du langage C# afin de l'utiliser dans la construction d'une application Web en ASP.
- ✦ Créer une application C# en mode console.
- ✦ Programmer en orienté-objet avec C#.
- ✦ Valider le fonctionnement d'une classe en C#.

INSTRUCTIONS

- Créer une solution Visual Studio nommée Tp1. Tous les programmes demandés sont des projets différents de la solution Tp1. Pour les premières applications, créez une classe pour vous permettre d'alléger le Main (pas de méthodes statiques sauf pour alléger le *Main*).
- Dans tous les cas où c'est pertinent, gérez les exceptions.
- Validez que votre code fonctionne bien avec un main complet (jeux de tests, toutes les méthodes) et pertinent et avec un test unitaire lorsque cela est demandé.
- Documentez, respectez la nomenclature et éviter la répétition de code.
- Vérifiez que vous avez tous les composants en complétant la grille de correction (colonne auto-évaluation).
- Quand ce n'est pas précisé, vous avez le choix du déroulement, de la réalisation ou de l'algorithme.
- Vous devez faire une démonstration du fonctionnement de l'ensemble de votre solution à l'enseignante.

PROJETS À DÉVELOPPER (COMPLEXITÉ CROISSANTE)

1. Créer un projet **Convertisseur** qui permet de convertir des degrés celsius en Kelvin et en Fahrenheit. Demandez la **saisie** à l'utilisateur et permettez-lui de saisir de nouveau s'il y a une erreur et de refaire une conversion s'il le désire.
2. Créer un projet **MiniCalcul** offrant les opérations élémentaires : +, -, /, * . Les opérations disponibles font partie d'une **énumération** et c'est ainsi que l'on pourra valider le choix de l'opérateur. L'utilisateur saisit son opérateur et s'il est valide on lui demande deux valeurs, si elles sont valides on fait l'opération et on affiche le résultat. Gérer les **exceptions** et redemander la saisie si l'utilisateur fait une erreur.
3. Créer un projet **Palindrome** qui permet de détecter si la chaîne saisie par l'utilisateur est un palindrome (mot qui peut être lu dans les deux sens). Si c'est le cas, reproduisez le palindrome dans un motif en utilisant une boucle **foreach** (lettres sur des lignes séparées et progressives et alternance **majuscules** et **minuscules**), par exemple :

K
 Ka
 KaY
 KaYa
 KaYaK

Demandez à l'utilisateur de saisir un mot, gérez les exceptions et redemandez si l'utilisateur veut faire un autre essai et **nettoyez** l'écran avant d'afficher le nouveau palindrome. Dans le cas où ce n'est pas un palindrome, on affiche que la chaîne ne constitue pas un palindrome.

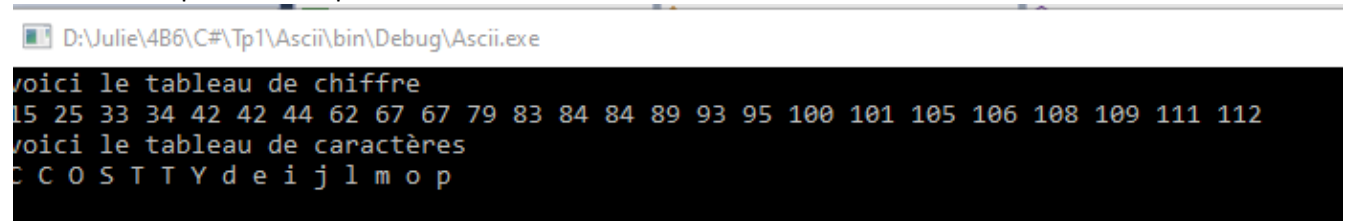
Faites un traitement itératif ou récursif pour vérifier si c'est un palindrome.

4. Créer un projet **Texte** qui demande à l'utilisateur une chaîne de caractères et affiche les statistiques de cette chaîne : nombre de majuscules avec un **Regex**, nombre de signes de ponctuation avec un **Regex**, nombre de caractères.
5. Créer un projet **ALEA** qui génère un tableau statique de 20 entiers aléatoirement dont les valeurs sont entre 1 et 100. Avec ce tableau, effectuez les opérations suivantes :
 - a. Afficher le tableau **trié** en ordre croissant,
 - b. Afficher uniquement les valeurs impaires trié en ordre croissant
 - c. Afficher la valeur maximale de ce tableau.

Une de vos méthodes doit utiliser la boucle **foreach**. Vous devez redéfinir la méthode **ToString** pour l'affichage du tableau. Le tout est présenté clairement à l'écran.

6. Créer un projet **ASCII** qui génère un tableau dynamique ArrayList de 25 entiers triés entre 15 et 125 et retourne dans une autre ArrayList de caractères, la lettre alphabétique correspondante (si elle existe) à la valeur numérique de la première collection. On affiche le tableau des valeurs et la conversion à l'écran. Portez attention à l'utilisation de constantes et à ne pas répéter de code. Chercher dans l'API si certaines méthodes d'objets peuvent vous être utiles (char, ArrayList, string, etc.).

Voici un exemple de sortie possible



```
D:\Julie\4B6\C#\Tp1\Ascii\bin\Debug\Ascii.exe
voici le tableau de chiffre
15 25 33 34 42 42 44 62 67 67 79 83 84 84 89 93 95 100 101 105 106 108 109 111 112
voici le tableau de caractères
C C O S T T Y d e i j l m o p
```

7. Créer un projet **CouleurS** qui permet de conserver les valeurs rvb d'une couleur et son nom. Par la suite, effectuez les traitements demandés à partir de cette structure.
 - a. Créer une couleur et l'enregistrer dans un vecteur (enregistrer avec des couleurs valides, avec une valeur rvb invalide, avec un nom null).
 - b. Afficher l'ensemble des couleurs du vecteur.
 - c. Chercher une couleur par son nom et afficher son code rvb. (chercher avec une couleur existante, inexistante, null).
 - d. Effacer une couleur.
 - e. Modifier une couleur (en fournissant la structure à modifier et la nouvelle structure).

Pour chaque opération, chercher des inexistants et des existants.

Pour la validation, on ne fait que valider si le code est entre 0 et 255, si ce n'est pas le cas on met 0 par défaut. Si le nom de la couleur est null on met une chaîne vide.

Créer un petit main pour tester l'ensemble de vos méthodes.)

Voici un exemple de sortie possible

```
Voici mes couleurs :
blanc : 255, 255, 255
rouge : 255, 0, 0
lime : 0, 255, 0
bleu : 0, 0, 255
noir : 0, 0, 0
jaune : 255, 255, 0
      : 255, 51, 255

Couleur correspondant à blanc : 255, 255, 255
Couleur correspondant à jaune : 255, 255, 0
Couleur correspondant à rose : Code inexistant
Effacer la couleur blanc : True
Effacer la couleur rose : False
Voici mes couleurs :
rouge : 255, 0, 0
lime : 0, 255, 0
bleu : 0, 0, 255
noir : 0, 0, 0
jaune : 255, 255, 0
      : 255, 51, 255

Réinitialiser
Voici mes couleurs :
blanc : 255, 255, 255
rouge : 255, 0, 0
lime : 0, 255, 0
bleu : 0, 0, 255
noir : 0, 0, 0
jaune : 255, 255, 0
```

PARTIE 2

Toujours dans votre main faites l'expérience suivante :

- f) Créez 2 couleurs c1 et c2 l'une avec ces valeurs : 128, 128, 128, "gris" et l'autre avec celles-ci : 255, 255, 255, "blanc".
- g) Maintenant, copiez la structure c2 dans c1 et constatez le résultat.
- h) Modifiez c1 pour les paramètres suivants : 255, 51, 255, "rose" puis faites afficher les valeurs des structures c1 et c2.

8. Créer un projet **CouleurO** et refaire le même travail qu'au numéro 7, mais avec un objet. Vous devez utiliser des getter/setter et redéfinir l'opérateur == pour comparer deux objets couleur identiques. Refaites aussi l'expérience de l'affectation et de la modification des couleurs c1 et c2.

Discuter des avantages et inconvénients des deux solutions et expliquez les différences constatées dans l'expérience de la structure et celle de l'objet.

9. Créer un projet **Produit**, qui contient une classe **Produit** et le **test unitaire** pour valider chacune des méthodes. Les propriétés sont :
- NoProduit, NomProduit, description, Prix, Quantité, QteRupture, Taxable.
- La propriété QtéRupture contient le nombre minimal de quantité en stock toléré.
- Le constructeur nécessite NoProduit, Nom. Les autres attributs sont à 0 ou à une chaîne vide ou à false et sont complétés grâce aux "getter".
- Tous les attributs ont un getter/setter. Le prix est entre 0 et 500, la quantité est entre 0 et 200, la qteRupture est entre 0 et 20% de la quantité.

On redéfinit les opérateurs d'égalité en lien avec le noProduit et les opérateurs < > en lien avec la quantité en stock - qté de rupture.

Un toString permet de retourner les attributs de l'objet dont la valeur est plus grande que 0 ou différente d'une chaîne vide.

Une méthode Total calcule la valeur en stock d'un produit (Quantite * Prix)

Créer un projet ProduitTest pour créer un test unitaire qui sert à vérifier chacune des méthodes publiques de votre classe Produit.

10. Créer un projet **Serialiser** qui contient une classe permettant de créer une liste de produits (5), puis :
- enregistrer en format texte tout le contenu de la liste
 - enregistrer en format XML la liste complète,
 - lire et afficher à l'écran le contenu d'un fichier texte
 - lire un fichier XML et initialiser la liste de produits et l'afficher ensuite.

Enregistrez les fichiers dans le répertoire courant.

Ayez une **seule méthode publique** pour lire, vous sélectionnez comment lire selon l'extension du fichier en majuscules ou minuscules. Vous enregistrez vos fichiers sous le nom **produits.xml** ou **produits.txt** sans demander à l'utilisateur.

Proposer un Main qui permet de vérifier que tout fonctionne bien. Prévoyez faire la preuve avec des données d'essai pertinentes. (exemple fichier absent, pas la bonne extension, etc.)

Bonus

Créer un projet **Température**, qui contient une collection Clé-Valeurs (Dictionary) des températures pour les dix premiers jours des mois de juin, juillet, août 2019 pour la ville de Québec. Voici, les températures pour ces trois mois :

Juin = { 19.7, 13.6, 14.9, 14.5, 19.9, 22.1, 22.1, 21.2, 27.3, 24.9 }

Juillet = { 24.4, 27.6, 28.4, 30.9, 29.4, 28.0, 25.1, 25.6, 28.1, 24.2 }

Août = { 27.7, 28.3, 23.5, 21.6, 24.3, 29.6, 26.8, 24.5, 22.9, 20.9 }

Le programme affiche dans un premier toutes les températures sans tenir compte du mois, puis la température la plus haute la température la plus basse et la moyenne des températures.

Ensuite on affiche les températures par mois et les statistiques pour chacun des mois (max, min, moyenne).

Ici, la difficulté est de parcourir le Dictionary qui contient comme valeur une liste de données de type double. Il n'y a pas qu'une seule façon, considérez la boucle foreach et les instructions Values, Keys, Value, Key, KeyValuePair de l'objet Dictionary.

À remettre

- Le travail se réalise en équipe sur 3 semaines incluant la démonstration. Le Tp est remis sur GIT et les dépôts doivent être documentés et alimentés par tous les équipiers.

Date de remise : **6 février 2020 – 17h**

420-355-LI Programmation Web

Grille de correction Tp1 – 10 %

Noms :

/100

Capacité 1	Excellent 100	Bien 85	Suffisant 60	Incomplet 30
Programmer des interfaces web interactifs 14 %	<p>Le code est bien structuré, propre et complet en respect avec les instructions fournies et les décisions prises dans l'analyse.</p> <p>Il est facile de se repérer dans l'application et elle est bien documentée.</p> <p>Le processus (thread) pour l'affichage des séquences est bien utilisé et fonctionnel.</p> <p>L'apparence de l'interface est conforme à la demande (grille, texte information, boutons) et au modèle annoncé.</p> <p>Les bonnes pratiques au niveau du positionnement des éléments dans les interfaces sont bien utilisées.</p> <p>Les classes sont bien définies et sont bien utilisées.</p> <p>Les outils de débogage sur un navigateur web sont utilisés de manière autonome.</p> <p>Des cas de tests pour les classes et pour les principales fonctions ont été fournis et appliqués.</p>	<p>Le code est bien structuré, propre en respect avec les instructions fournies et les décisions prises dans l'analyse.</p> <p>Il est généralement facile de se repérer dans l'application et elle est bien documentée.</p> <p>Le processus (thread) pour l'affichage des séquences est utilisé et généralement fonctionnel.</p> <p>L'apparence de l'interface est conforme à la demande (grille, texte information, boutons) et au modèle annoncé.</p> <p>Les bonnes pratiques au niveau du positionnement des éléments dans les interfaces sont utilisées.</p> <p>Les classes sont généralement bien définies et sont bien utilisées.</p> <p>Les outils de débogage sur un navigateur web sont utilisés généralement de manière autonome.</p>	<p>Le code est généralement bien structuré et propre en respect, la plupart du temps, avec les instructions fournies et les décisions prises dans l'analyse.</p> <p>Le processus (thread) pour l'affichage des séquences peut comporter certaines difficultés.</p> <p>L'apparence de l'interface est généralement conforme à la demande (grille, texte information, boutons) et au modèle annoncé, mais il peut y avoir des manques, mais qui n'entraînent pas un dysfonctionnement.</p> <p>Les classes sont présentes mais à améliorer.</p> <p>Les outils de débogage sur un navigateur web sont utilisés.</p> <p>Le travail est fait avec de l'aide occasionnelle.</p> <p>Les dépôts sur GIT sont à améliorer dans leur fréquence, leur documentation et leur clarté.</p>	<p>Le code est difficile à lire et comporte des lacunes au niveau de sa structure.</p> <p>Le processus (thread) pour l'affichage des séquences ne fonctionne pas.</p> <p>L'apparence de l'interface comporte plusieurs lacunes.</p> <p>Les classes ne sont pas utilisées.</p> <p>Les outils de débogage sur un navigateur web sont rarement utilisés.</p> <p>Le travail est fait avec de l'aide constante.</p> <p>Les dépôts sur GIT sont minimaux et peu documentés..</p>

	<p>Le travail est fait de façon autonome.</p> <p>Les dépôts sur GIT sont faits de façon régulière, parfaitement repérables et les différentes versions sont clairement identifiées et documentées.</p>	<p>Des cas de tests pour les classes et pour les principales fonctions ont été fournis et appliqués la plupart du temps.</p> <p>Le travail est généralement fait de façon autonome.</p> <p>Les dépôts sur GIT sont faits de façon régulière, repérables et généralement documentées.</p>		
Capacité 3	Excellent 100	Bien 85	Suffisant 60	Incomplet 30
Appliquer des mesures de sécurité dans une application web 4 %	<p>Des traces complètes de tests sont laissées dans le code pour les classes et les principales fonctions de l'application et en lien avec les cas de tests.</p> <p>L'application fonctionne parfaitement sur les principaux navigateurs.</p>	<p>Des traces de tests sont laissées dans le code pour les classes et les principales fonctions de l'application et en lien avec les cas de tests.</p> <p>L'application fonctionne parfaitement sur la plupart des principaux navigateurs</p>	<p>Quelques traces de tests sont laissées dans le code pour les classes et les principales fonctions de l'application.</p> <p>L'application fonctionne généralement bien sur la plupart des principaux navigateurs</p>	<p>Peu ou pas de traces de tests.</p> <p>L'application ne fonctionne que sur un navigateur.</p>