

Cours de Programmation

Travaux Pratiques 2

ITII PACA - A&M ParisTech 2010-2011



Ruding LOU
ruding.lou@ensam.fr

Découverte de l'environnement de développement Visual Studio &

Programmation impérative en VB/Visual Studio

Tout au long des exercices proposés, prenez soin de lire et de comprendre les éventuelles erreurs de compilation ou d'exécution. Sauvez chaque exercice dans un projet (ou un module VB) différent, et familiarisez-vous avec les différentes fonctionnalités du logiciel.

TP 2.1 Système de gestion des livres

Concevoir un système de gestion des livres. Le système propose un menu avec plusieurs items de fonctionnalité. Chaque fois l'utilisateur entre le numéro pour accéder à une fonctionnalité, une fois la fonctionnalité se termine le menu du système réapparaît. Les fonctionnalités sont ordonnées comme suivant :

- 1) Ajouter un auteur dans le système en registrant les informations nécessaires
- 2) Ajouter un livre dans le système en enregistrant les informations nécessaires
- 3) Supprimer un auteur depuis le système
- 4) Supprimer un livre depuis le système
- 5) Afficher tous les livres d'un auteur spécifique
- 6) Afficher tous les livres publiés à partir d'une date de publication spécifique
- 7) Modifier les informations d'un auteur
- 8) Modifier les informations d'un livre
- 9) Exporter toutes les données du système vers un fichier du format spécifique sur le disque dur.
- 10) Importer des données depuis un fichier du format spécifique sur le disque dur

Spécifications:

Créer une structure de données (un type) qui s'appelle **Auteur**. Cette structure doit pouvoir stocker des informations suivantes :

- Identifiant d'auteur (généralisé automatiquement, distinct par rapport tous les autres auteurs)
- Nom de la personne
- Prénom de la personne
- Date de naissance de la personne
- Nationalité de la personne
- Livres écrits par cette personne et existant dans ce système

Créer une structure de données (un type) qui s'appelle **Livre**. Cette structure doit pouvoir stocker des informations suivantes :

Ajouter un livre en enregistrant les informations suivantes :

- Identifiant du livre (généralisé automatiquement, distinct par rapport tous les autres livres)
- Titre du livre
- Date de publication
- auteurs du livre
- Prix du livre
- Nombre d'exemplaires

Le système ne doit pas permettre d'avoir deux personnes ayant des informations identiques : nom, prénom et date de naissance. Le système ne doit pas permettre d'avoir deux livres ayant des informations identiques : titre et date de publication.

Quand on supprime un livre depuis le système il faut mettre à jour les informations des livres pour les auteurs concernés. Quand tous les livres d'un auteur sont supprimés depuis le système, l'auteur doit être aussi supprimé depuis le système. Le système ne doit pas supprimer un auteur depuis le système tant qu'il y a son(ses) livre(s) stocké(s) dans le système.

Pour afficher les livres le format doit être comme ceci :

```
* * * * *
*  Introduction à la programmation  *
*                                  *
*                                GATES Bill *
*                                1999-09-09 *
*  $ 19.50                        *
*  en stock: 30 exemplaires        *
* * * * *
```

Outils possible à utiliser (Consulter la documentation en appuyant F1 pour savoir en détails):

- Ouverture du fichier en mode lecture (`FileSystem.OpenTextFileReader`)
- Ouverture/Création du fichier en mode écriture (`FileSystem.OpenTextFileWriter`)

TP 2.2 jeu de voiture

Suite de l'exercice "course de chevaux" dans cet exercice il s'agit de réaliser une simulation d'un jeu de voiture dans le quel il y a un affichage dynamique et aussi une interaction de l'utilisateur en temps réel.

On imagine une scène suivante :

Une route de largeur définie et de longueur infinie.

Sur la route il y a des obstacles dont la largeur est plus petite que la largeur de route et assez petite pour que les voitures puissent les contourner.

L'utilisateur avance sur la route avec une voiture avec laquelle il peut se déplacer vers la gauche ou la droite et il peut aussi accélérer ou freiner.

Le challenge est que la voiture ne doit pas cogner les obstacles.

Réalisation de la simulation :

Pour la simulation

- On se focalise sur la zone devant la voiture d'une distance bien définie (par exemple 20).
- On simplifie les obstacles par des lignes avec une largeur limitée à une valeur qui est raisonnablement inférieure à la largeur de route.
- On simplifie la voiture par une lettre "H" qui finalement occupe une case d'affichage sur la route.
- On fait défiler les lignes (obstacles) pour simuler l'avancement de la voiture.
- On laisse l'utilisateur utiliser les quatre touches directionnelles ← → (pour diriger la voiture à gauche/droite) et ↑ ↓ (pour accélérer/freiner la voiture).
- On considère que la voiture entre en collision avec l'obstacle si la lettre "H" occupe la case où une ligne d'obstacle y passe aussi.

Pour l'affichage dynamique :

On affiche 20 lignes pour la route devant la voiture et une ligne pour la voiture. Parmi les vingt lignes on affiche des espaces là où il n'y a pas d'obstacle et des symboles (par exemple "*") pour là où il y a une partie d'un obstacle. L'affichage des vingt lignes se rafraichit pour faire défiler les lignes d'obstacles.

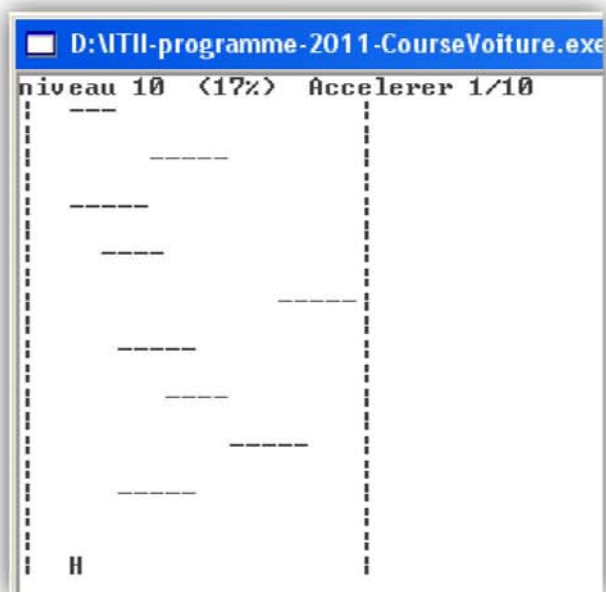
On affiche la 21^{ème} ligne avec des espaces et une lettre (par exemple "H") sur la case représentant la position de la voiture.

Outils possibles à utiliser (Consulter la documentation en appuyant F1 pour savoir en détails) :

- ☞ Lire une touche `Console.ReadKey()`
- ☞ Création des threads pour exécuter des tâches parallèlement `Thread`, `Thread.start(...)`

Voici deux images de capture d'écran au deux moments.

Moment 1 (17^{ème} seconde):



Moment 2 (29^{ème} seconde):

