Chapitre 6 :
Exemples de programmation

UE Projet Informatique 2 -
Programmation C#

Intérêt du chapitre

- Comment programmer des choses simples?
  - Trier
  - Accéder aux fichiers
  - Créer des jeux

Objectifs

- Utiliser les techniques fondamentales de programmation dans des cas courants
  - Mettre en œuvre les algorithmes standards de tri
  - Crypter des données sauvegardées
  - Accéder aux fichiers
  - Créer des jeux

Sommaire

Organisation de la mémoire

Généralités

- Au sein d'une application, la mémoire disponible est découpée en deux grandes zones :
  - La pile (en anglais stack)
  - Le tas (en anglais heap)

Organisation de la mémoire

La pile

- La pile est une zone mémoire gérée automatiquement par le compilateur.
- C'est la zone par défaut d'allocation des variables
  - compilateur va automatiquement réserver de la mémoire dans la pile pour cette dernière
  - limitée en taille
  - il est important qu'il puisse libérer cette mémoire le plus tôt possible

Organisation de la mémoire

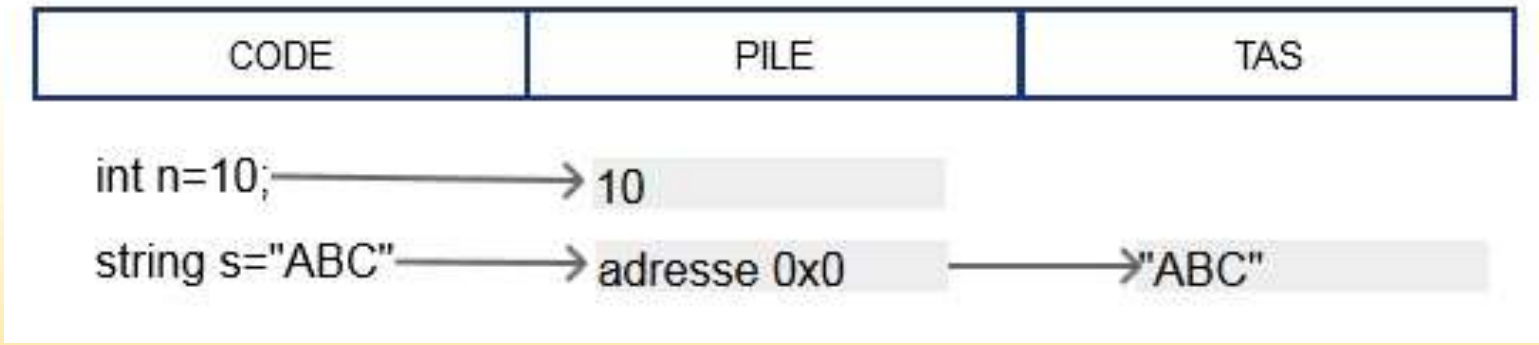
Le tas

- l'utilisation du tas doit être faite explicitement par le programmeur par l'utilisation de l'allocation dynamique
  - Cela peut être fait manuellement avec des pointeurs nus ou par des objets (instruction new)
  - Cette zone est virtuellement illimitée
    - tant qu'il y a de la mémoire de libre, on peut créer des objets dedans

Organisation de la mémoire

Exemple

- L'espace mémoire utilisé par une donnée de type valeur est affecté à la pile
- Pour les types références les données réelles du type sont stockées dans le tas et un pointeur vers ces données est stocké dans la pile



Exemple de tri

Généralités

- Un algorithme de tri permet d'organiser une collection d'objets selon une relation d'ordre déterminée
- La collection à trier est souvent donnée sous forme de tableau ou sous forme de liste
  - afin de permettre l'accès direct aux différents éléments de la collection

Exemple de tri

Tri par sélection ou tri par extraction

	8
	5
	2
	6
	9
	3
	1
	4
	0
	7

```
procédure tri_selection(par reference tableau d'entiers t)
    n ← longueur_tableau(t)
    pour i de 0 à n - 2
        min ← i
        pour j de i + 1 à n - 1
            si t[j] < t[min], alors min ← j
        fin pour
        si min ≠ i, alors permuter( t[i],t[min])
    fin pour
fin procédure
```

Travaux pratiques

Programme Tri par sélection

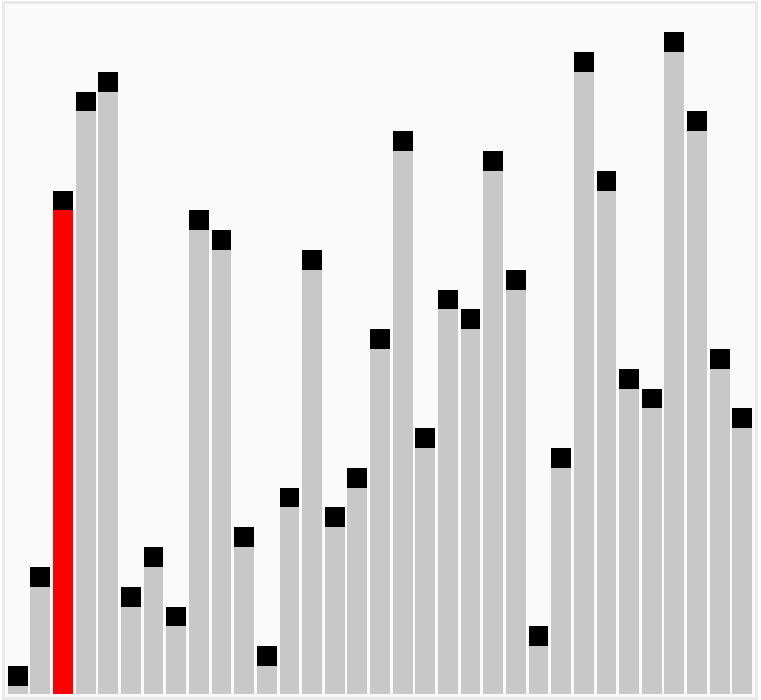
- Ecrire un programme qui demande en entrée un liste de nombres entiers séparés par des espaces et qui retourne la liste de nombre triée
- Astuces
  - Utilisé Split()

```
Saisissez un suite de nombre separé par des espaces :
12 8 5
Voici les nombres dans l'ordre
5 8 12
```

Exemple de tri

Tri à bulles

```
tri_bulles_optimise(par reference Tableau d'entiers T)
    pour i allant de taille de T - 1 à 1
        tableau_trié := vrai
        pour j allant de 0 à i - 1
            si T[j+1] < T[j]
                permuter(T[j+1], T[j])
                tableau_trié := faux
            fin si
        fin pour
        si tableau_trié
            sortir procédure
        fin si
    fin pour
```



Travaux pratiques

Programme Tri bulle

- Ecrire un programme qui prend en ligne de commande un liste de nombres entiers séparés par des espaces et qui retourne la liste de nombre triée
- Astuces
  - Utilisez args

```
static int Main(string[] args)
{
```

  - Ensuite Projet -> Propriétés -> Déboguer->Arguments

Cryptographie

cryptage XOR

- Espace de nom : System.Security.Cryptography
- système de cryptage basique
  - facile à implémenter
  - La fonction de cryptage est identique à la fonction de décryptage
- Il tient son nom de l'opérateur logique XOR "OU exclusif".

Lettres	M	E	S	S	A	G	E
Codes ASCII	77	69	83	83	65	71	69
Binaire	01001101	01000101	01010011	01010011	01000001	01000111	01000101
Message en binaire	01001101	01000101	01010011	01010011	01000001	01000111	01000101
Clé en binaire (répétée si nécessaire)	01000011	01001100	01000101	01000011	01001100	01000101	01000011
Message crypté en binaire	**00001110**	**00001001**	**00010110**	**00010000**	**00001101**	**00000010**	**00000110**

Travaux pratiques

Programme de modification de mot de passe

- On doit écrire un programme
  - qui demande votre ancien mot de passe
  - Si il correspond au mot enregistré dans les paramètres
    - Vous demande le nouveau mot de passe et l'enregistre
- Construire l'algorithme
- Ecrire une fonction crypto()
- Utilisez le cryptage xor pour l'enregistrement du mot de passe
  - Utilisez Properties.Settings.Default

Accès aux fichiers textes

Généralités

- Les principales fonctions sont définies dans System.IO:
  - Classe File, Directory
    - Fournit des méthodes statiques pour gérer les fichiers
    - Copy(); Exist()
  - StreamReader(String)
    - Initialise une nouvelle instance de la classe StreamReader pour le nom de fichier spécifié
    - Read(); ReadLine(); ReadToEnd();
    - Close();

Travaux pratiques

Programme afficher un fichier texte

- On doit écrire un programme qui prend le nom d'un fichier en argument de la ligne de commande et qui affiche son contenu

- Construire l'algorithme
- Ecrire le programme

Accès aux fichiers binaires

Généralités

- FileStream (au lieu de StreamReader)
  - Méthode Read(), ReadByte(), Write(), Seek()
  - Propriété Position

Travaux pratiques

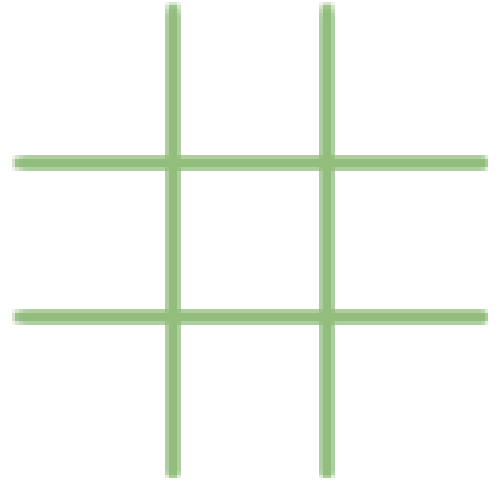
Programme calculer le checksum d'un fichier binaire

- On doit écrire un programme qui prend le nom d'un fichier en argument de la ligne de commande et qui affiche la valeur de son checksum
  - Construire l'algorithme
  - Ecrire le programme
  - Intégrer la gestion des erreurs (fichier non trouve, etc)
- Astuces
  - Utiliser MD5.ComputeHash de System.Security.Cryptography

Exemples de jeux simples

Le morpion

- Le jeu du morpion ou tic-tac-toe
  - consiste à aligner des ronds ou des croix en ligne, colonne ou diagonale sur un plateau de 3x3 cases
  - Le premier joueur qui aligne ses pions a gagné



Travaux pratiques

Programme le jeu de MORPION

- Quelques indications
  - Le tableau dispose de deux dimensions, 3x3
  - Il faut au sein d'une boucle permuter les joueurs.
  - Si une position est déjà occupée, il faut de nouveau poser la question
  - Après chaque coup, il faut vérifier toutes les lignes, colonnes et diagonales
  - Si une ligne, colonne, diagonale est complète, on sort de la boucle
  - En cas de victoire, il faut indiquer quel pion (x, o) a gagné
  - Il faut gérer le match nul : neuf tours et personne n'a gagné
  - Copier le code source depuis le support de cours
  - Modifier le code source afin de jouer contre l'ordinateur
    - L'ordinateur prend la place du joueur x et cherche une position libre dans le tableau de façon aléatoire

Travaux de recherche

Chercher le fonctionnement des éléments suivants

- Tri par insertion
- Tri rapide
- Cryptographie DES
- Jeu des tours de Hanoï

Conclusion…

Il y a tant à coder…