

T.P. N°1 en Programmation évènementielle

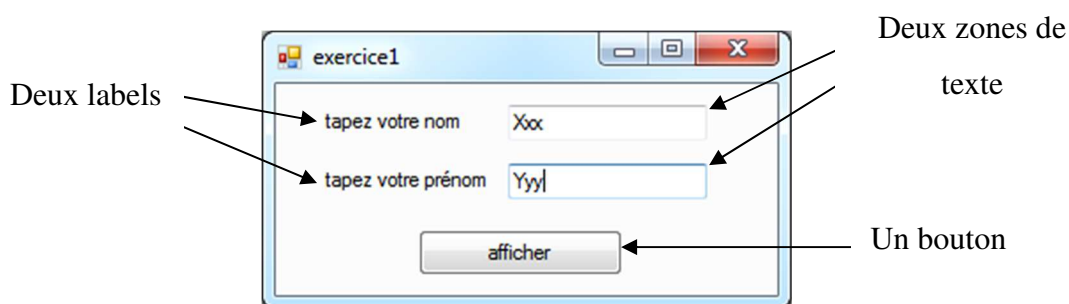
Enseignante : M^{me} Soumaya GHORBEL YANGUI

Auditoire: Deuxième année licence

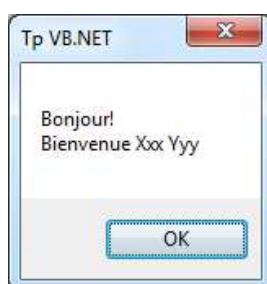
N.B. Chaque application que vous créez doit être personnalisé (couleur, taille de la police, arrière plan...)

Exercice n°1

A l'aide de Visual Basic .NET, créer un formulaire contenant deux zones de texte, deux labels et un bouton « afficher », comme le montre la figure ci-dessous.



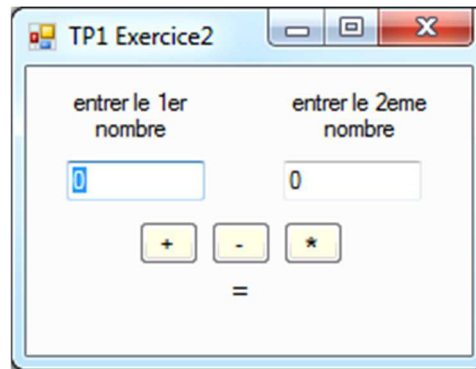
- 1) En cliquant sur le bouton « afficher », un message sera affiché en appelant la méthode ***MessageBox.Show(...)***. Le message contient la chaîne « Bonjour ! Bienvenue » qui sera concaténée avec le nom et le prénom tapés dans les deux zones de texte.



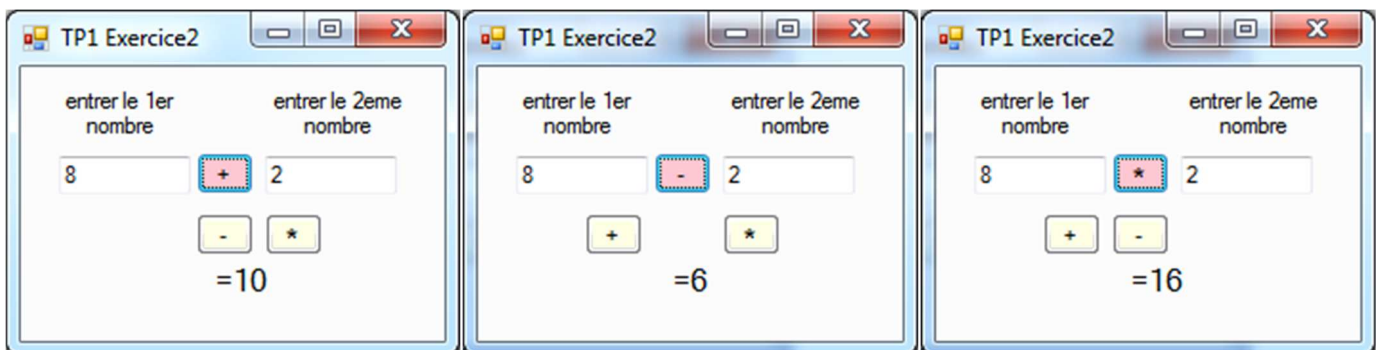
- 2) Ajouter un autre bouton « effacer » permettant d'effacer les contenus des zones de texte.

Exercice n°2

Créer une application permettant de calculer la somme, la soustraction et le produit de deux nombres réels saisis dans deux zones de texte. Le résultat sera affiché dans un label. Le formulaire comporte trois boutons pour effectuer ces opérations, voir la figure ci-dessous.



- 1) Le clic sur l'un des boutons permet de modifier sa position et sa couleur d'arrière plan, ainsi il permet d'afficher le résultat correspondant, voir les illustrations suivantes.



- 2) Considérons le cas d'addition des deux nombres. Modifier l'application de telle façon lorsqu'on modifie l'une des valeurs des zones de texte, le résultat sera calculé et affiché sans avoir cliqué sur le bouton.

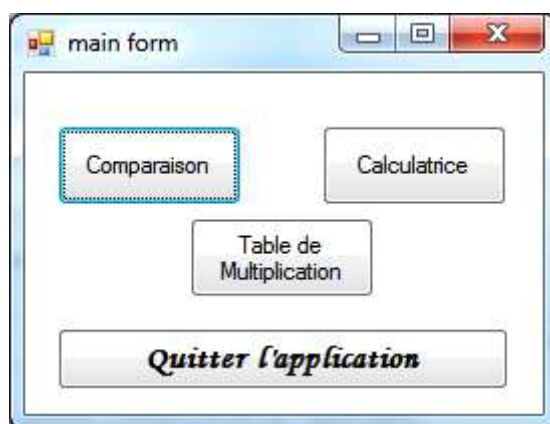
T.P. N°2 en Programmation évènementielle

Enseignante : M^{me} Soumaya GHORBEL YANGUI

Auditoire: Deuxième année licence

Exercice n°1

Créer une application vb.net comportant quatre formulaires dont le formulaire principal est représenté par la figure suivante :



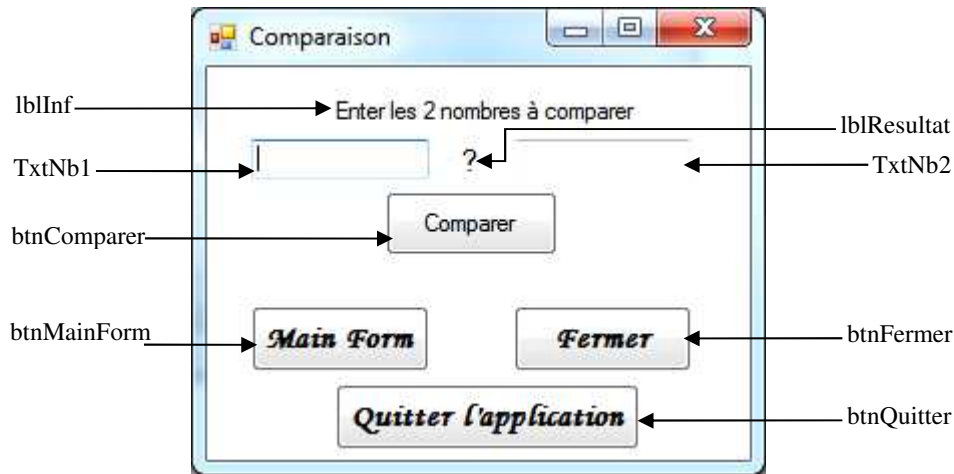
Ce formulaire, intitulé *main form* et identifié par le nom *MainForm*, comporte ces quatre boutons :


- Le bouton *Comparaison* (identifié par *btnComparer*) :
 - Le clic sur ce bouton permet d'afficher un nouvel formulaire intitulé *Comparaison*.
- Le bouton *Calculatrice* (identifié par *btnCalculer*) :
 - Le clic sur ce bouton permet d'afficher un nouvel formulaire intitulé *Calculatrice*.
- Le bouton *Table de Multiplication* (identifié par *btnMultiplier*) :
 - Le clic sur ce bouton permet d'afficher un formulaire intitulé *Table de Multiplication*.
- Le bouton *Quitter l'application* (identifié par *btnQuitter*) :
 - Le clic sur ce bouton permet d'arrêter l'exécution de l'application.

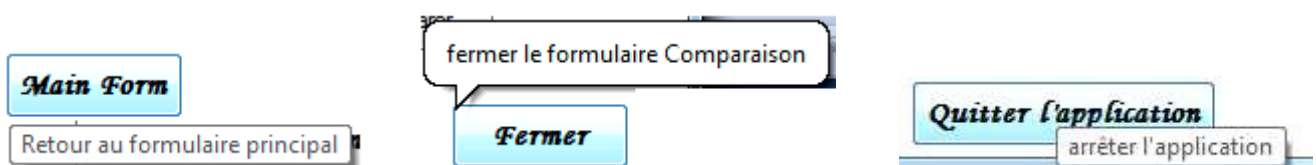
La description de ces différents formulaires est détaillée dans les exercices suivants.

Exercice n°2

Considérons le formulaire *Comparaison* représenté ci-dessous et dont les noms de ses différents composants sont indiqués dans la figure suivante :



- 1) Ce formulaire comporte deux zones de texte pour la saisie de deux nombres réels. Le clic sur le bouton *Comparer* (identifié par *btnComparer*) déclenche la comparaison entre ces deux nombres saisis. Ainsi, le texte du label *lblResultat* « ? » sera remplacé par « < », « > » ou « = ».
- 2) Le formulaire comporte aussi trois autres boutons :
 - Le bouton *Main Form* : le clic sur ce bouton permet d'afficher le formulaire principal et de le mettre au premier plan.
 - Le bouton *fermer* : le clic sur ce bouton permet de fermer le formulaire courant.
 - Le bouton *Quitter l'application* : le clic sur ce bouton permet d'arrêter l'exécution de l'application.
- 3) Utiliser l'outil  **ToolTip** qui sert à afficher des informations (des info-bulles) lorsque l'utilisateur passe le pointeur sur un contrôle associé, pour informer des rôles des différents boutons présents dans le formulaire *Comparaison*.



Exercice n°3

Considérons le formulaire *Table de Multiplication* représenté par la figure suivante :

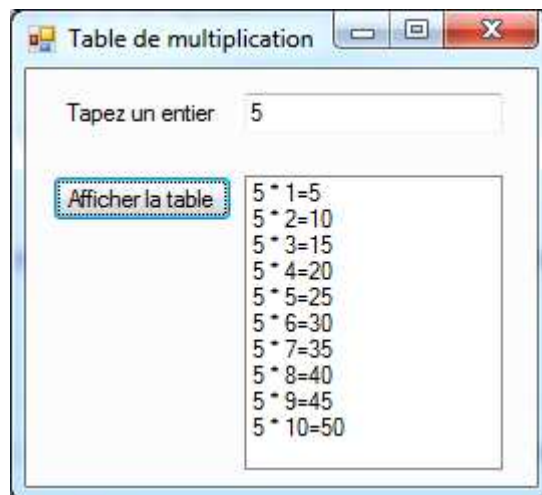


Table de multiplication

Tapez un entier: 5

Afficher la table

- 5 * 1=5
- 5 * 2=10
- 5 * 3=15
- 5 * 4=20
- 5 * 5=25
- 5 * 6=30
- 5 * 7=35
- 5 * 8=40
- 5 * 9=45
- 5 * 10=50

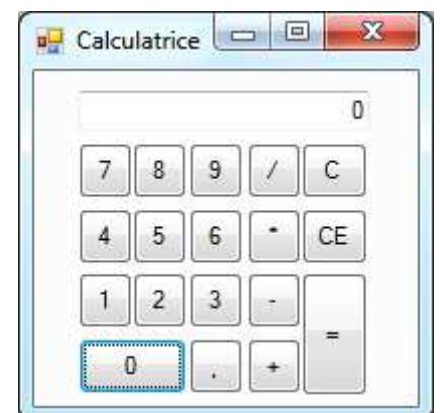
Ce formulaire permet de calculer et d'afficher la table de multiplication d'un entier donné saisi dans une zone de texte. Il comporte un bouton pour déclencher le calcul et l'affichage de la table de multiplication de l'entier dans une liste de choix *ListBox*.

1. Ajouter le code nécessaire pour remplir la liste avec les valeurs appropriées en cliquant sur le bouton "Afficher la table".
2. Modifier le code pour avoir l'affichage de la table de multiplication dans une liste vide à chaque clic sur le bouton "Afficher la table" (pas d'affichage multiple de plusieurs tables de multiplication dans une même liste).
3. Ajouter le code nécessaire pour afficher la valeur 1 dans la zone du texte ainsi que sa table de multiplication dans la liste au moment du chargement du formulaire de l'application.
4. Améliorer l'application en affichant la nouvelle table de multiplication pour chaque changement de la valeur dans la zone du texte, sans cliquer sur le bouton "Afficher la table".

Exercice n°4

Considérons le formulaire *Calculatrice* représenté par la figure ci-contre. Ce formulaire, permettant de simuler une calculatrice, contient une zone de texte pour saisir les opérandes de types réel, quatre boutons représentant les opérations possibles tels que (+, -, *, /), des boutons pour insérer des chiffres, un bouton intitulé « CE » pour annuler l'opérande saisi, un bouton intitulé « C » permet de supprimer le dernier caractère saisi avec le clavier et un bouton pour insérer le symbole de ponctuation. Le résultat sera affiché dans le même champ des opérandes.

N.B. : la zone de texte peut avoir au plus une virgule.



Calculatrice

0


7 8 9 / C

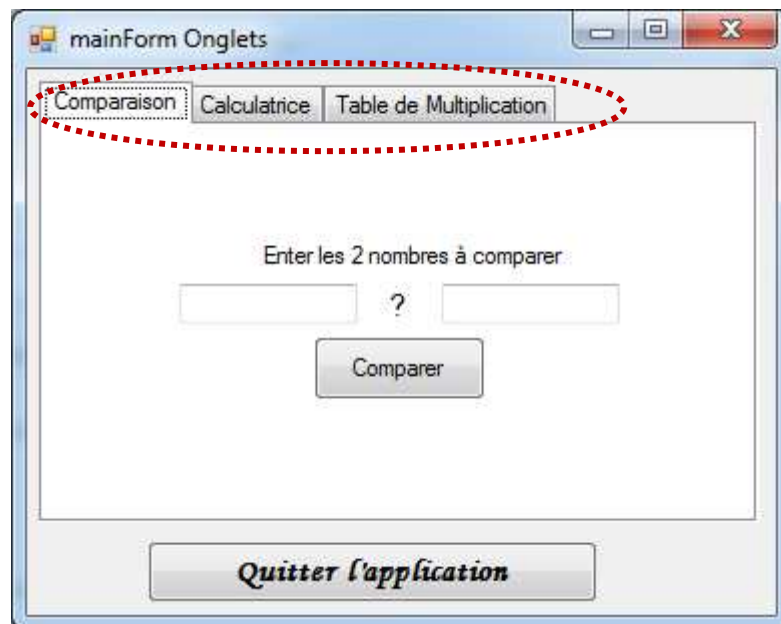
4 5 6 * CE

1 2 3 -

0 . + =

Exercice n°5

Créer une autre application vb.net similaire à ce qui est créée précédemment mais représentée par un seul formulaire en utilisant les onglets grâce à l'outil  **TabControl**.



mainForm Onglets

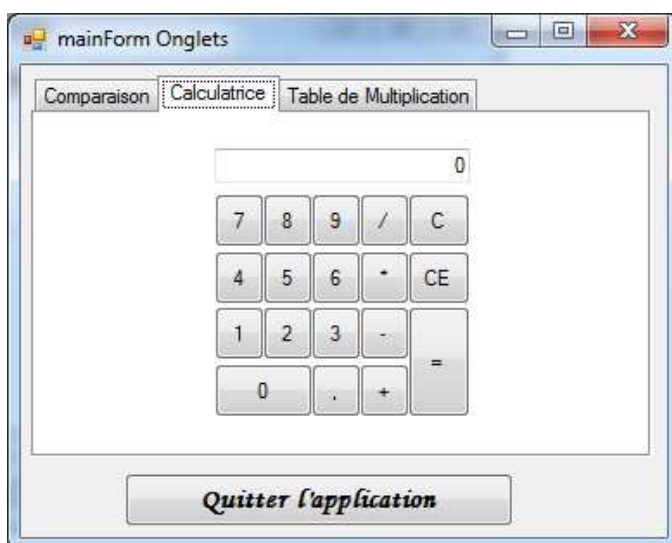
Comparaison Calculatrice Table de Multiplication

Enter les 2 nombres à comparer

?

Comparer

Quitter l'application



mainForm Onglets

Comparaison **Calculatrice** Table de Multiplication

0

7 8 9 / C

4 5 6 * CE

1 2 3 - =

0 . +

Quitter l'application



mainForm Onglets

Comparaison Calculatrice **Table de Multiplication**

Tapez un entier

Afficher la table

Quitter l'application

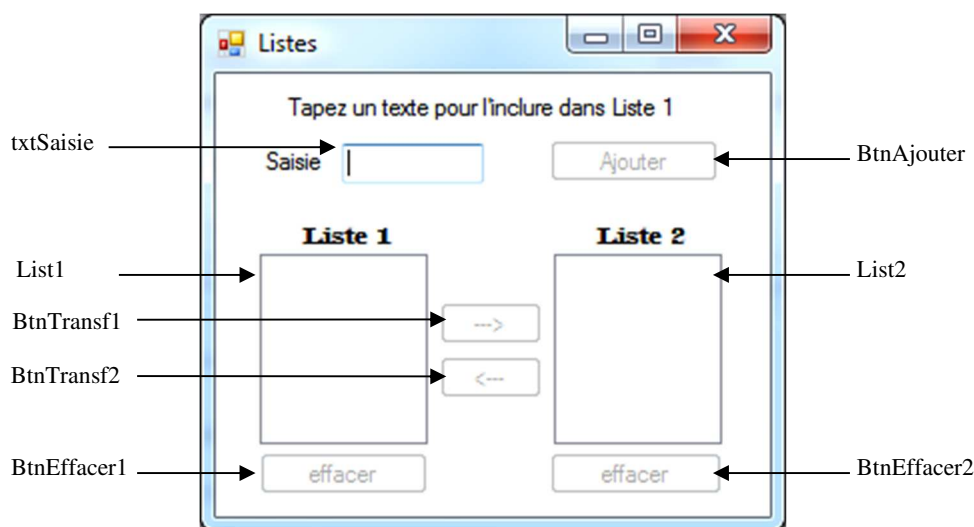
T.P. N°3 en Programmation évènementielle

Enseignante : M^{me} Soumaya GHORBEL YANGUI

Auditoire: Deuxième année licence

Exercice n°1

Construire l'interface suivante qui comporte les composants de type : textbox, button, label et listbox.

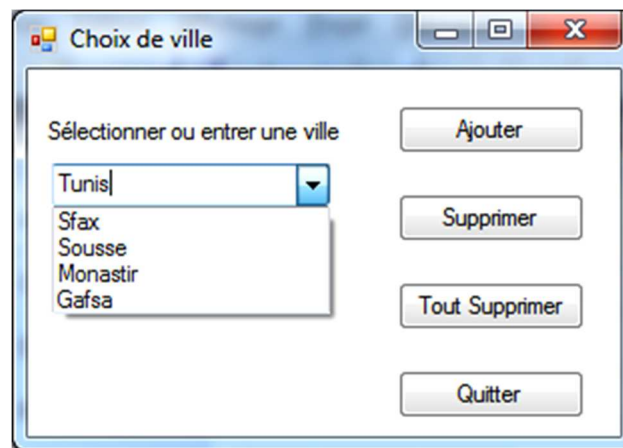


Fonctionnement

1. Lors du chargement du formulaire, tous les boutons sont désactivés.
2. Le bouton *Ajouter* sera actif lorsque la zone de texte est non vide.
3. Le clic sur le bouton *Ajouter* permet d'ajouter le contenu de la zone de texte dans la liste 1. Le champ de saisie est alors vidé et l'utilisateur peut ajouter un nouvel élément.
4. L'utilisateur peut transférer des éléments d'une liste à l'autre en sélectionnant l'élément à transférer dans l'une des listes et en choisissant le bouton de transfert adéquat → ou ←. L'élément transféré est ajouté à la fin de la liste de destination et enlevé de la liste source.
 - a. le bouton de transfert de la liste 1 vers la liste 2 n'est activé que s'il y a un élément sélectionné dans la liste 1
 - b. le bouton de transfert de la liste 2 vers la liste 1 n'est activé que s'il y a un élément sélectionné dans la liste 2
5. Les boutons d'effacement des listes 1 et 2 ne sont activés que si la liste à effacer contient des éléments.
6. L'utilisateur peut double-cliquer sur un élément de la liste 1. Cet élément est alors transféré dans la boîte de saisie pour modification et enlevé de la liste 1.

Exercice n°2

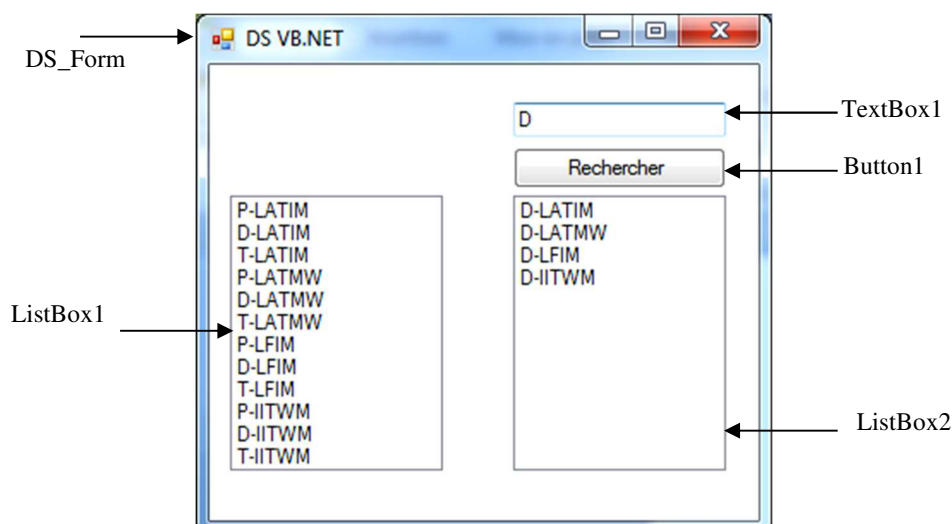
L'interface ci-dessous montre une liste déroulante (Combobox) dans laquelle une liste de villes est stockée.



1. Le clic sur le bouton "Ajouter " permet d'ajouter le nom de la ville saisie à la liste. L'ajout n'est effectué que si le nom de la ville ne figure pas dans le Combobox. L'ajout se fait en appelant une fonction « AjoutVille(v) », qui permet d'ajouter la ville "v" à la liste. Un message « la ville est ajoutée avec succès » sera affiché si la ville est ajoutée, et un message « la ville existe déjà » sera affiché si la ville existe déjà dans la liste.
2. Le clic sur le bouton "Supprimer" permet de supprimer la ville sélectionnée après confirmation de l'utilisateur : un messagebox s'affiche demandant la confirmation de l'utilisateur avant d'effectuer la suppression.
3. Le bouton "Tout Supprimer" permet de vider la liste.

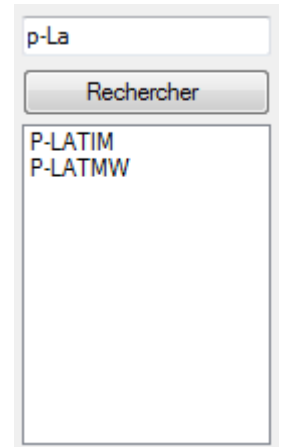
Exercice n°3

Soit l'interface d'un formulaire donnée par la figure suivante et qui comporte une zone de texte, un bouton et deux listes.



Donner le code nécessaire pour la réponse aux questions suivantes :

1. Lors du chargement du formulaire le bouton «Button1 » est désactivé et la liste «ListBox2 » est invisible.
2. Lorsqu'on tape sur la zone de texte (le contenu est modifié) :
 - a. Le bouton devient actif et la liste «ListBox2 » devient visible.
 - b. Une recherche se déclenche pour trouver dans la liste «ListBox1» les éléments qui commencent avec les premières lettres tapées dans la zone de texte ; le résultat de la recherche sera affiché dans la liste «ListBox2».
 - c. La recherche ne doit pas être sensible à la casse (le même résultat de la recherche pour des lettres tapées en minuscules ou en majuscules dans la zone de texte).
 - d. La liste «ListBox2» doit être vidée avant de fournir le résultat d'une nouvelle recherche.



Exercice n°4

Un usager de La Poste veut envoyer un courrier. On lui fournit les tarifs (partiels pour simplifier). Il veut connaître, en fonction du poids et du type de son courrier (lettre ou colis), quel est le prix à payer pour l'envoi de celui-ci, en service rapide et en service économique.

Une lettre est caractérisée par : son poids (en Kg), le mode d'expédition (express ou normal), son adresse de destination et son format ("A3" ou "A4"). Un colis est caractérisé par : son poids, le mode d'expédition, son adresse de destination et son volume (en cm³).

Voici les règles utilisées pour affranchir le courrier :

1. en mode d'expédition normal :

- le montant nécessaire pour affranchir une lettre dépend de son format et de son poids :

montant = tarif de base + 1.0 * poids,

où le tarif de base pour une lettre "A4" est de 2.50, et 3.50 pour une lettre "A3"

- le montant nécessaire pour affranchir un colis dépend de son poids et de son volume :

Formule : montant = 0.25 * volume + poids * 1.0

2. en mode d'expédition express : les montants précédents sont doublés, quelque soit le type de courrier.

Travail demandé :

1) Construire l'interface suivante sachant que :

- Les deux groupbox « lettre » et « colis » sont désactivés au moment du chargement du formulaire.
- Chaque groupbox, « lettre » ou « colis », sera activé si la case à cocher correspondante est cochée.

Choisir le type de votre courrier(s) ☒ Lettre ☐ Colis

Lettre

Poids Kg

Format

Mode d'expédition ☒ Normal ☐ Express

Adresse de Destination

Colis

Poids Kg

Volume

Mode d'expédition ☐ Normal ☒ Express

Adresse de Destination

Courriers à expédier

	N° courrier	Type	Poids	Mode d'expédition	Adresse	Tarif
▶	1	Lettre	0,5	Normal	Sfax...	4
	2	Colis	40	Express	Tunis...	50

Total TTC = 54 Dinars

- 2) Implémenter les événements click des deux boutons « Ajouter », qui permettront de calculer le tarif du courrier choisi et d'ajouter les détails du courrier ainsi que le prix de l'expédition dans le DataGridView. Désactiver le groupbox correspondant après le calcul et décocher les checkbox.
- 3) Implémenter l'événement click du bouton « Calculer Tarif » qui permet de calculer le tarif total des courriers à envoyer et d'afficher le montant total au dessous de DataGridView.

Exercice n°5

- 1) Créer une classe « lettre » et une classe « colis » représentant respectivement les lettres et les colis détaillés dans l'exercice précédent.
- 2) Créer un tableau de type « lettre » et un autre tableau de type « colis ». Ces tableaux sont des variables globales déclarés dans un module standard.
- 3) Modifier le code des deux boutons Ajouter de l'exercice précédent pour pouvoir ajouter les données saisies dans les tableaux correspondants créés.
- 4) Ajouter à l'application des boutons **charger**, **modifier** et **supprimer** qui permettent respectivement de charger les données à partir des tableaux, de modifier et de supprimer des données.
- 5) Ajouter un bouton **chercher** qui permet de chercher le plus lourd courrier.

T.P. N°4 en Programmation évènementielle

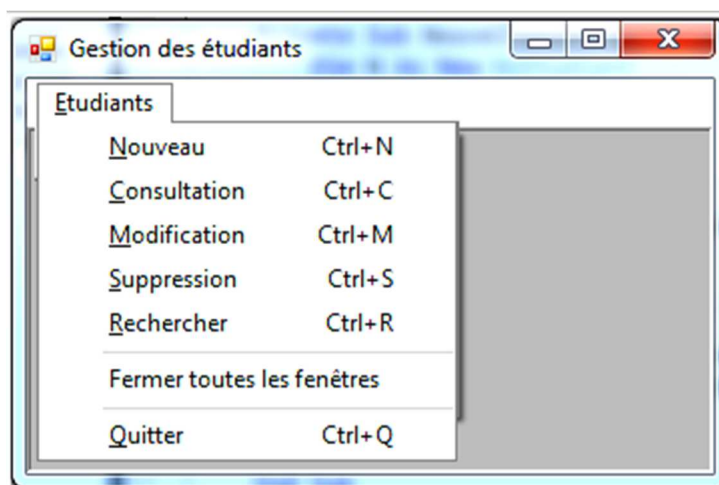
Enseignante : M^{me} Soumaya GHORBEL YANGUI

Auditoire : Deuxième année licence


Partie A

On souhaite créer une application qui gère la liste des étudiants au sein d'une institution universitaire. L'application comporte un formulaire parent dans lequel il y a un menu principal pour la gestion des étudiants nommée « **Etudiants** » regroupant les sous menus :

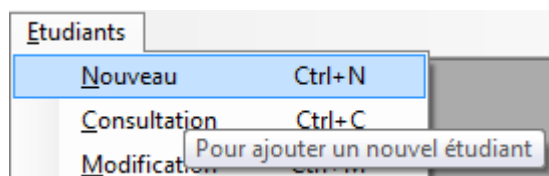
- **Nouveau** : pour ajouter un nouvel étudiant
- **Consultation** : pour consulter la liste présente des étudiants
- **Modification** : pour modifier les données d'un étudiant
- **Suppression** : pour supprimer un étudiant de la liste.
- **Rechercher** : pour chercher des étudiants selon des critères donnés.
- **Fermer toutes les fenêtres** : pour fermer les fenêtres ouvertes dans l'application.
- **Quitter** : pour quitter l'application.



1. Créer l'interface du formulaire parent donnée par la figure ci-dessus :

- ☞ La Propriété **Form.IsMdiContainer** permet d'obtenir ou de définir une valeur indiquant si le formulaire est un conteneur de formulaires enfants d'interface multidocument (MDI).
- ☞ L'outil  **MenuStrip** de la boîte à outils sert à fournir un système de menus d'un formulaire.

2. Décrire l'objectif de chaque sous menu en ajoutant des info-bulles correspondantes



3. En cliquant sur le sous menu **Nouveau**, un nouveau formulaire « **NvEtudiant** », permettant d'ajouter un nouvel étudiant à la liste, s'ouvre en tant qu'un formulaire enfant du formulaire parent déjà créé.

- a. Créer l'interface du formulaire « **NvEtudiant** » donnée par la figure ci-dessous :

Création d'un nouvel Etudiant

Matricule :

CIN:

Nom:

Prénom:

Ville:

Email:

Inscription:

Filière:

Niveau:

Groupe:

Date de Naissance :

décembre 2015						
lun.	mar.	mer.	jeu.	ven.	sam.	dim.
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

Aujourd'hui : 03/12/2015

Enregistrer

Annuler

- b. Ecrire le code nécessaire pour associer un formulaire enfant « **NvEtudiant** » au formulaire parent créé.

☞ La propriété **Form.MdiParent** permet d'obtenir ou de définir le formulaire parent MDI en cours de ce formulaire.

4. En cliquant sur le bouton **Enregistrer** du formulaire « **NvEtudiant** », les données du nouvel étudiant seront vérifiées, puis validées et enregistrées :

- a. Créer une structure de données appropriée pour l'enregistrement de la liste des étudiants.
 - b. Créer une fonction qui permet de chercher un étudiant, étant donnée son matricule, dans la liste des étudiants. La fonction retourne la position de l'étudiant dans la liste s'il existe et elle retourne -1 dans le cas contraire.
 - ☞ Déclarer les variables et les méthodes globales dans un module standard.
 - c. Le champ *CIN* doit avoir exactement 8 chiffres.
 - d. Le champ *E-mail* doit avoir le caractère « @ ».
 - e. Le bouton *Enregistrer* est actif si les champs *Matricule*, *Nom* et *Prénom* ne sont pas vides.
 - f. L'enregistrement d'un nouvel étudiant ne peut s'effectuer que si son matricule n'est pas attribuée à un autre étudiant présent dans la liste.

5. Le bouton **Annuler** permet de vider tous les champs du formulaire ainsi de le fermer.

6. En cliquant sur le sous menu **Consultation**, le formulaire enfant « **CsltEtudiants** » permettant la consultation de la liste des étudiants s'ouvre.

- a. Créer l'interface du formulaire « **CsltEtudiants** » donnée par la figure ci-dessous, qui comporte la composante « **DataGridView** » pour afficher la liste des étudiants.

Consultation de la liste des Etudiants

	Matricule	CIN	Nom	Prénom	Ville	Email	Filière	Niv
▶	mat1	00110011	nom1	prenom1	sfax	nom1.prenom1@...	LF	
	mat2	00220022	nom2	prenom2	sfax	nom1.prenom1@...	LATIM	

- b. La composante « DataGridView » sera en mode lecture seule et permet d'afficher les différents attributs des étudiants présents dans la liste lors du chargement du formulaire.
 - c. En effectuant un double clic sur l'en-tête d'une ligne de la composante « DataGridView », le formulaire « MdfEtudiant » s'ouvre en tant que formulaire enfant permettant la modification des données de l'étudiant sélectionné à partir de la composante « DataGridView ». Le même résultat sera affiché lors du clic sur le sous menu **Modification**.
 - ☞ L'événement **DataGridView.RowHeaderMouseDoubleClick** se produit lors d'un double-clic sur un en-tête de ligne.
 - d. Le formulaire « MdfEtudiant » est similaire au formulaire « NvEtudiant », mais avec une autre couleur d'arrière-plan. Le champ *Matricule* est en mode lecture seule ; c'est l'unique champ qui ne peut pas être modifié.
 - e. Lors du chargement du formulaire « MdfEtudiant » les données, de l'étudiant sélectionné dans le « DataGridView », sont transférées vers les différents champs présents dans ce formulaire.
 - f. Le clic sur le bouton **Enregistrer** du formulaire « MdfEtudiant » permet d'enregistrer les modifications correspondantes et de fermer ce formulaire.
7. Le clic sur le sous menu **Suppression** permet de supprimer l'étudiant sélectionné dans la composante « DataGridView » du formulaire « CsltEtudiants ». Un message sera affiché permettant de valider ou non la suppression.
 8. Le clic sur le sous menu **Rechercher** permet d'ouvrir le formulaire enfant « RechEtudiants » qui comporte une composante « DataGridView » pour afficher la liste des étudiants vérifiant certains critères. Le formulaire comporte aussi des champs pour la saisie de ces critères ainsi que deux boutons **Rechercher** et **Actualiser**.
 - a. Créer ce formulaire représenté par la figure suivante :

Rechercher des Etudiants

Rechercher des étudiants selon les critères saisis

Matricule : mat

CIN:

Nom: nom

Prénom:

Ville:

Email:

Inscription:

Filière:



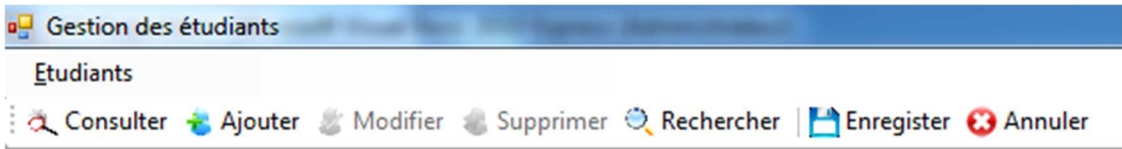
Niveau:

Groupe:

Il y a 2 étudiants

	Matricule	CIN	Nom	Prénom	Ville	Email	
	mat1	00110011	nom1	prenom1	sfax	nom1.prenom1@...	L
▶	mat2	00220022	nom2	prenom2	sfax	nom1.prenom1@...	L

- b. Lors du chargement de ce formulaire le « DataGridView » comporte l'ensemble de tous les étudiants.

- c. En cliquant sur le bouton Rechercher  , une recherche se déclenche permettant d'afficher la liste des étudiants vérifiant les critères saisis dans la composante « DataGridView ».
 - d. En effectuant un double clic sur l'en-tête d'une ligne de la composante « DataGridView », le formulaire « MdfEtudiant» s'ouvre permettant la modification des données de l'étudiant sélectionné à partir de la composante « DataGridView » de ce formulaire. Le même résultat sera affiché lors du clic sur le sous menu **Modification**.
9. Actualiser l'affichage des données dans le « DataGridView » après une opération d'ajout, de modification ou de suppression.
10. Enrichir votre application en ajoutant une barre d'outils à l'aide de l'outil  **ToolStrip** .
- a. Cette barre d'outils comporte des boutons facilitant l'accès à différentes opérations.
- 
- b. Les boutons **Modifier** et **Supprimer** ne sont actifs que lorsqu'il ya des éléments sélectionnées dans la composante « DataGridView » du formulaire « CsltEtudiants» ou du formulaire « RechEtudiants» permettant respectivement la modification et la suppression de l'étudiant sélectionné.
 - c. Les boutons **Enregistrer** et **Annuler** ne sont visibles que lors d'un ajout ou d'une modification d'un étudiant (les formulaires « MdfEtudiant» et « NvEtudiant »sont actifs).
11. Ajouter un menu contextuel au « DataGridView ».
12. Ajouter un écran de démarrage à votre application, qui s'ouvre lors du lancement de l'application et se ferme automatiquement après 3 secondes.

Partie B

1. Créer une base de données qui contient une table Etudiant représentant l'étudiant avec les champs cités précédemment.
2. Etablir une connexion à la base de données en mode connecté puis en mode déconnecté.