

IHM - Programmation événementielle :

Visual Basic .NET

Documents autorisés.

Durée : 2 heures 30.

Sujet : réalisation d'une application d'inscription à un examen.

Il s'agit de recueillir les inscriptions à un examen administratif national et d'en faire le bilan. Le cahier des charges est le suivant :

— chaque candidat doit saisir ses coordonnées, indiquer sa région d'inscription, choisir les épreuves qu'il présentera à l'écrit et à l'oral et éventuellement une épreuve facultative orale ;

— les 10 régions sont :

<i>Auvergne</i>	<i>Bordelais</i>	<i>Bourgogne</i>	<i>Bretagne</i>	<i>Corse</i>
<i>Nord</i>	<i>Normandie</i>	<i>Paris</i>	<i>Poitou</i>	<i>Roussillon</i>

— une même matière ne peut être présentée qu'une seule fois à l'écrit ou à l'oral, ou en épreuve facultative ;

— les 12 matières suivantes peuvent être proposées à l'écrit, mais seules celles accompagnées d'un * dans la liste suivante peuvent être présentées à l'oral :

<i>Allemand*</i>	<i>Anglais*</i>	<i>Astronomie*</i>	<i>Biologie</i>	<i>Espagnol*</i>	<i>Français</i>
<i>Géographie</i>	<i>Géologie</i>	<i>Histoire*</i>	<i>Italien*</i>	<i>Mathématiques</i>	<i>Philosophie</i>

— les 3 matières suivantes ne sont proposées qu'à l'oral :

Arts Musique Sport

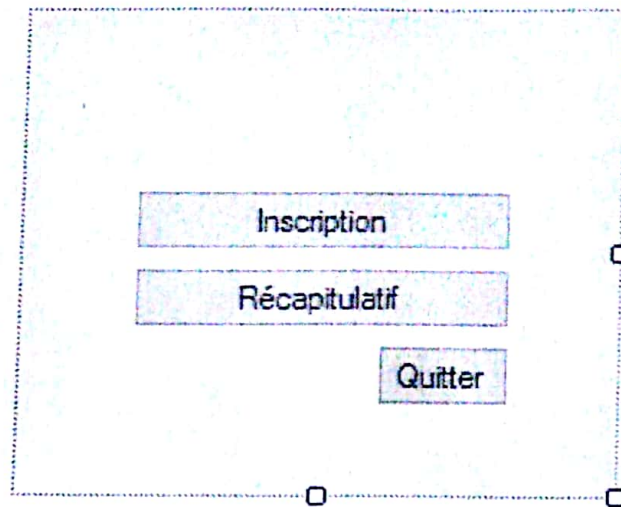
— chaque candidat doit présenter obligatoirement 3 matières différentes à l'écrit et 2 autres à l'oral. L'épreuve facultative est à choisir en dernier parmi toutes les matières non sélectionnées à l'écrit comme à l'oral.

Vous développerez le code d'une telle application, d'après les interfaces fournies et en vous appuyant sur les noms des contrôles proposés par cet énoncé ; pour chaque question, vous fournirez le code permettant d'obtenir le comportement décrit par la question, c'est à dire le ou les gestionnaires d'événements pour les interfaces, ou structure fonctions et procédures en ce qui concerne le module.

En ce qui concerne l'entête des gestionnaires d'événements, vous ne vous souciez pas du type des paramètres *sender* et *e*, et ne mentionnez ces paramètres que lorsqu'il sont utiles à la gestion de l'événement.

Rien ne vous oblige à traiter les questions dans l'ordre de l'énoncé.

1 Le formulaire d'Accueil *frmAcc*



Au démarrage du programme, le **formulaire Principal** de l'application *frmAcc* proposera 3 boutons à l'utilisateur :

- **Q(uestion)1.** le **Bouton Inscription** *btnInscr* : masque le **formulaire** principal et affiche le **formulaire** Inscription *frmInsc*;
- **Q2.** le **Bouton Récapitulatif** *btnRecap* : affiche le **formulaire** Récapitulatif *frmRecap* sans masquer le **formulaire** principal;
- **Q3.** le **Bouton Quitter** *btnQuit* : met fin à l'application.

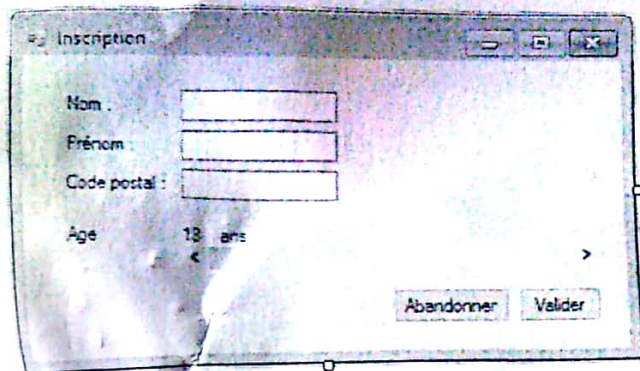
2 Le module *Candidats*

Tout au long de son exécution, l'application devra retenir les informations concernant les inscriptions des candidats à l'examen. Votre module devra proposer :

- **Q4.** une **structure de données** permettant de mémoriser l'intégralité des informations nécessaires à l'enregistrement de l'**inscription** d'un candidat, puis un **tableau** pour y enregistrer les différentes inscriptions (l'**indice d'une inscription dans ce tableau servira de numéro au candidat**);
- **Q5.** une **fonction** renvoyant le nombre d'inscriptions enregistrées;
- **Q6.** une **fonction** permettant d'enregistrer une nouvelle inscription, cette fonction renvoie l'indice auquel le candidat a été enregistré dans le tableau;
- **Q7.** une **fonction** pour obtenir une inscription à partir de son indice dans le tableau;
- **Q8.** une **procédure** **Main()** qui initialisera le tableau et démarrera le formulaire principal.

3 Inscription d'un candidat *frmInscr* et *frmOpt*

Une inscription sera réalisée à partir de 2 feuilles : l'une pour la saisie des renseignements du candidat et l'autre pour le choix des épreuves.



Renseignements à saisir sur le formulaire d'inscription *frmInscr* :

- nom : **TextBox** *txtNom* alphabétique, 15 caractères au maximum ;
- prénom : **TextBox** *txtPre* : alphabétique, 20 caractères au maximum ;
- code postal : **TextBox** *txtPost* numérique, 5 chiffres, les 2 premiers sont différents de 0 ;
- age : **ScrollBar** *scrAge* de 18 à 55 ans, à choisir en nombre d'année.

Ce formulaire doit proposer les contrôles suivants :

- **Q9.** un bouton **Abandonner** *btnAbd* permet, à tout moment, d'annuler l'inscription en cours et de revenir au formulaire principal *frmAcc* ;
- **Q10.** la longueur des zones de saisie et les paramètres de la *ScrollBar* *scrAge* sont fixés au chargement du formulaire ;
- **Q11.** le contrôle du type des champs (alphabétique ou numérique) est réalisé via les événement **KeyPress** ;
- **Q12.** le déplacement de la *ScrollBar* modifiera l'affichage de l'âge sélectionné dans le **Label** *lblAge* ;
- **Q13.** un bouton **Valider** *btnVld* permet, après contrôle et si le formulaire est bien rempli, le passage au second formulaire pour le choix des épreuves (tous les champs sont obligatoires, il n'est pas nécessaire de signaler les champs vide lors de la validation) ;

Choix des options (temps restant : 41 sec)

Epreuves écrites : 0
 CheckBox1
 CheckBox2
 CheckBox3
 CheckBox4
 CheckBox5
 CheckBox6
 CheckBox7
 CheckBox8
 CheckBox9
 CheckBox10
 CheckBox11
 CheckBox12

Epreuves orales : 0
 CheckBox13
 CheckBox14
 CheckBox15
 CheckBox16
 CheckBox17
 CheckBox18
 CheckBox19
 CheckBox20
 CheckBox21

Nom : Rosal
 Prénom : Julien
 Région d'examen : Nord

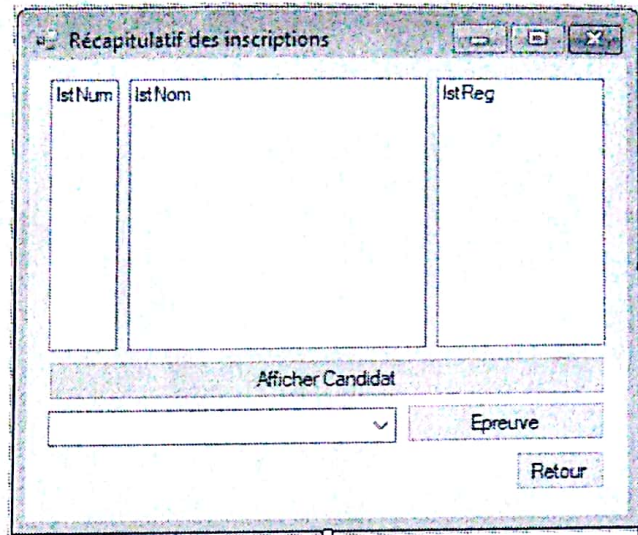
Option facultative : oui non

Enregistrement

Choix de la région de passage et des épreuves :

- **Q14.** le nom et prénom du candidat seront rappelés par des labels *lblNom* et *lblPre*. les régions qui organisent ces épreuves seront proposées par une **ComboBox** *cmbRegion* où apparaît initialement dans la zone de texte la première de ces régions. Il ne doit pas être possible de saisir une région qui n'est pas proposée par l'application. Ces trois contrôles sont initialisés au chargement du formulaire :
- **Q15.** pour les matières obligatoires, le choix est assisté par 2 séries de **CheckBox** regroupées dans deux **Panels** distincts, correspondants aux matières disponibles d'une part à l'écrit *pnlEcrit* et d'autre part à l'oral *pnlOral* (Vous ne vous occuperez pas de charger les intitulés des matières). Le candidat ne devra pas pouvoir sélectionner plus de 3 matières à l'écrit ni plus de 2 matières à l'oral, un message signalera le dépassement. Au dessus de chaque liste, le candidat verra s'afficher, mis à jour à chaque sélection ou dé-sélection, le nombre de choix restants via les Labels *lblEcrit* et *lblOral*.
De plus, à chaque choix de matières à l'écrit, cette matière doit devenir indisponible à l'oral si elle en faisait partie et redevenir disponible si ce choix d'écrit est abandonné. Il en est de même pour le choix à l'oral ;
Le **Panel** *pnlValid* sera disponible si et seulement l'utilisateur a sélectionné exactement 3 matières à l'écrit et 2 à l'oral ;
- **Q16.** deux **RadioButtons** (*rbOui/rbNon*) disponibles dès la fin du choix des matières obligatoires lui permettent de choisir s'il veut ou non s'inscrire à une épreuve facultative :
 - le choix *oui* désactivera les **Panels** de choix d'épreuves, puis fera apparaître une **ComboBox** *cmbFac* qui proposera toutes les matières non encore sélectionnées, et rendra indisponibles les cadres des choix obligatoires ;
 - le choix *non* videra et fera disparaître la **ComboBox**, puis activera les **Panels** de choix d'épreuves.
- **Q17.** Un **Bouton** permet de valider l'inscription. Une **MsgBox** récapitulera l'inscription (nom, prénom, code postal, âge, région pour l'examen, matières choisies) et demandera confirmation. La validation provoquera l'enregistrement du candidat dans le module et affichera le numéro attribué au candidat, puis retournera au formulaire d'accueil.
- **Q18.** les options devront être choisies dans un temps limité à 1 min. Le temps restant apparaît dans le titre de la feuille, actualisé chaque seconde. Ce temps écoulé, il y a retour au formulaire d'accueil après notification par **MsgBox** sans confirmation.

4 Le formulaire Récapitulatif *frmRecap*



Ce formulaire comportera 3 **ListBox** synchronisées avec les informations des inscriptions présentées comme suit : le numéro d'inscription (*lstNum*), la concaténation du nom et du prénom (*lstNom*) et la région d'inscription (*lstReg*).

- **Q19.** Le contenu de ces **ListBox** est chargé lors du lancement de ce formulaire (l'affichage des numéros de candidats ainsi que des régions doivent correspondre pour une même ligne) ;
- **Q20.** la sélection doit être synchronisée sur les trois listes. Par exemple, lors de la sélection d'un numéro de candidat, son nom et prénom ainsi que la région qu'il a choisie doivent se sélectionner dans les listes correspondantes.
- **Q21.** un **Bouton** Candidat *btnCand* permet d'afficher un **MsgBox** récapitulant les informations du candidat sélectionné par les **ListBox** ainsi que ses différents choix d'épreuve.
- **Q22.** une **ComboBox** *cmbEpr* ainsi qu'un **Bouton** Épreuve *btnEpr* permettent de sélectionner une matière et d'afficher, par **MsgBox** la liste des candidats (nom, prénom, code postal et région d'examen) ayant choisi cette épreuve. Pour chaque candidat, vous préciserez si l'épreuve est écrite, orale ou facultative. Vous ne vous occuperez pas de charger la liste des intitulés de matière dans la **ComboBox** ;
- **Q23.** un **Bouton** *btnRet* permet de fermer la fenêtre et de revenir au formulaire d'accueil *frmAcc*

IHM - Programmation événementielle :

Visual Basic .NET

*Documents autorisés.
Durée : 2 heures.*

Objectif : réalisation d'une application de gestion des commandes de petits déjeuners du lendemain pour un hôtel.

Cette application permettra :

- d'être un client, pour effectuer une prise de commande;
- d'être l'hôtelier, pour visualiser le bilan des commandes.

Cet hôtel a 6 étages numérotés de 1 à 6, et 16 chambres par étage numérotées de 1 à 16. Le numéro d'une chambre est composé du numéro d'étage suivi du numéro de la chambre de l'étage (par exemple 215 est le numéro de la chambre 15 du 2^{ème} étage, et la chambre 230 n'existe pas).

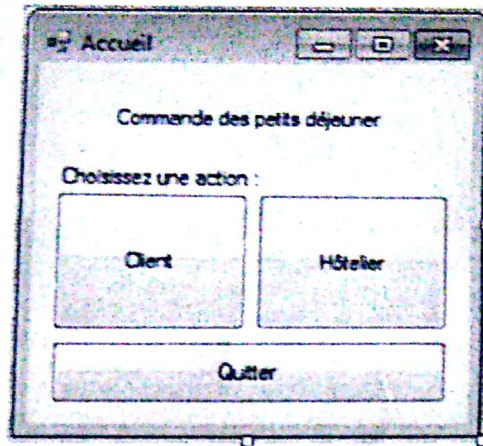
Vous développerez le code d'une telle application, d'après les interfaces fournies et en vous appuyant sur les noms des contrôles proposés par cet énoncé.

Pour chaque question, vous fournirez le code permettant d'obtenir le comportement décrit par la question, c'est à dire le ou les gestionnaires d'événements pour les interfaces, ou structure fonctions et procédures en ce qui concerne le module.

En ce qui concerne l'entête des gestionnaires d'événements, vous ne vous souciez pas du type des paramètres *sender* et *e*, et ne mentionnez ces paramètres que lorsqu'il sont utiles à la gestion de l'événement.

Rien ne vous oblige à traiter les questions dans l'ordre de l'énoncé.

1 Le formulaire d'Accueil *frmAccueil*



Au démarrage du programme, le **formulaire principal** de l'application *frmAccueil* proposera 3 boutons à l'utilisateur :

- Q(uestion)1. le bouton **Client** *btnClient* : masque le formulaire principal et affiche le formulaire *Client frmClient*;
- Q2. le bouton **Hôtelier** *btnHotel* : masque le formulaire principal et affiche le formulaire *Hôtelier frmHotel*;
- Q3. le bouton **Quitter** *btnQuit* : met fin à l'application.

Application .exit()

2 Le module *Commandes*

Tout au long de son exécution, l'application devra retenir les informations concernant les commandes de petits déjeuners des clients de l'hôtel. Un module devra proposer :

- Q4. une **structure de données** permettant de mémoriser l'intégralité des informations nécessaires¹ à l'enregistrement de la **commande** d'une chambre, ainsi qu'un **tableau** d'éléments de cette structure ;
- Q5. une fonction renvoyant le nombre de commandes enregistrées dans le tableau ;
- Q6. une procédure permettant d'enregistrer une nouvelle commande ;
- Q7. une fonction renvoyant l'indice dans le tableau d'une commande identifiée par un numéro de chambre, -1 si elle n'existe pas ;
- Q8. une fonction pour obtenir une commande à partir de son indice dans le tableau ;
- Q9. une procédure **Main()** qui initialisera le tableau et lancera le formulaire principal.

Application .begin() & .showForm

3 Le formulaire Client *frmClient*

Pour chaque prise de commande, on commencera par obtenir le numéro de la chambre, à valider par un bouton :

- Q10. au Clic sur le bouton **Rechercher** *btnRech*, en fonction du numéro de chambre saisi dans le champ **TextBox** *txtChbr* :
 - le numéro saisi n'est pas valide ou vide, ou la commande est déjà prise : le signaler par **MsgBox**, puis revenir au formulaire principal **frmAccueil** ;
 - le numéro saisi existe, et la commande n'est pas encore prise : le **Panel** *pnlCommande* devient actif par propriété **Enabled** et permet la prise de la commande :

Lorsque le **Panel** *pnlCommande* est activé, il est demandé à l'utilisateur d'indiquer :

- son nom, par la **TextBox** *txtNom* ;
- son prénom, par la **TextBox** *txtPrenom* ;
- si le petit déjeuner est pris en chambre ou en salle, par 2 **RadioButtons** *rbtChbr* et *rbtSalle* ;
- le choix d'une boisson, par une **ComboBox** *cmbBoisson* déjà remplie (vous n'avez pas à la remplir), proposant "café", "thé", "chocolat" ou "jus d'orange" ;
- un choix de viennoiseries par un ensemble de **CheckBox**, regroupées dans un **GroupBox** *grpVien*, proposant "pain chocolat", "croissant", "toasts", "muffin" ou "chausson".

Le formulaire doit alors réagir de la façon suivante :

- **Q11.** un bouton **Abandonner** *btnAbandon* permet, à tout moment, d'annuler la commande en cours et de revenir au **formulaire principal** *frmAccueil*;
- **Q12.** les **CheckBox** pour le choix des viennoiseries sont regroupées dans un **GroupBox** *grpVien*. Facultatives, on ne doit pas pouvoir en cocher plus de 2 ;
- **Q13.** le nom et le prénom saisis par l'utilisateur doivent être strictement alphabétiques, ceci doit être vérifié pendant la saisie en empêchant l'utilisateur d'entrer des caractères non alphabétiques ;.
- **Q14.** Si le bouton *rbtChbr* est coché, une **ScrollBar** *scrHDej* ainsi qu'un **Label** *lblHDej* apparaissent par propriété **Visible**. Il permettent à l'utilisateur de choisir l'heure à laquelle le petit déjeuner lui sera servi. La **ScrollBar** *scrHDej* permet de sélectionner un horaire entre 6 heures et 9 heures, à 15 minutes près. Le **Label** *lblHDej* affiche l'heure sélectionnée. Par défaut, il proposera l'horaire 7 H 30.
Si l'utilisateur coche le bouton *rbtSalle*, la **ScrollBar** *scrHDej* et le **Label** *lblHDej* disparaissent ;

Client

Recherche ou prise d'une commande par numéro de chambre 300 Rechercher

Informations client :

Nom : Le Castor

Prénom : Totot

Service du petit déjeuner :

en salle

en chambre : 7h30

Composition du petit déjeuner :

Boisson : Café

Viennoiseries

Pan au chocolat

Croissant

Toasts

Muffin

Chausson aux pommes

Enregistrer Abandonner

- **Q15.** au Clic sur le bouton **Enregistrer** *btnEnr*, si les tous renseignements ont été saisis (nom, prénom et choix d'une boisson), une **MsgBox** récapitulera les informations et demandera une confirmation. Si la commande est validée, les informations sont enregistrées et l'application retourne au **formulaire principal** *frmAccueil*. Dans le cas contraire, les champs incorrects sont signalés par une couleur de fond rouge et la commande se poursuit. La couleur de fond d'origine des champs de saisie sera restaurée dès qu'une nouvelle saisie sera détectée.

4 Le formulaire hôtelier *frmHotel*

Le formulaire hôtelier *frmHotel* permet de récapituler les commandes à prévoir. Ce formulaire affiche, lorsqu'il est activé :

- **Q16.** un **Label** *lblNbSalle* affiche le nombre de repas à prévoir en salle, puis dans trois **List-Box**, *lstChbr*, *lstHeure* et *lstCom* synchronisées, un récapitulatif final des petits déjeuners pris en chambre. Chaque commande s'affichera sur une ligne, son numéro de chambre dans la liste *lstChbr*, l'heure de service dans la liste *lstHeure*, puis le détail de la commande (boisson et viennoiseries) dans la liste *lstCom*. Ces commandes seront triées selon l'heure du service ;
- **Q17.** ces trois listes seront synchronisées pour la sélection : la sélection d'un item dans une liste provoquera la sélection des items correspondants à la même ligne (même commande) dans les deux autres listes.

Recapitulatif final - 10:00 avant fermeture

Prévisions pour le service de demain

Nombre de petits déjeuners à prévoir en salle : 0

Planning prévisionnel des services en chambre :

Chambre	Horaire	Détail de la commande
lstChbr	lstHeure	lstCom

Quitter

deje

Une fois ces informations affichées, ce formulaire se fermera :

- **Q18.** automatiquement et sans confirmation, passé un délai de 10 minutes, par un **Timer** *tmrQuit*. Le temps restant avant fermeture de l'application sera affiché dans la barre de titre de ce formulaire ;
- **Q19.** lors d'un clic sur le bouton **Quitter** *btnQuit*.