

Direction Régionale Tensift Atlantique Marrakech

Examen de Fin de Module, Formation Initiale
Année 2014 -2015

Filière : TDI

Niveau : TS

Durée : 1h30

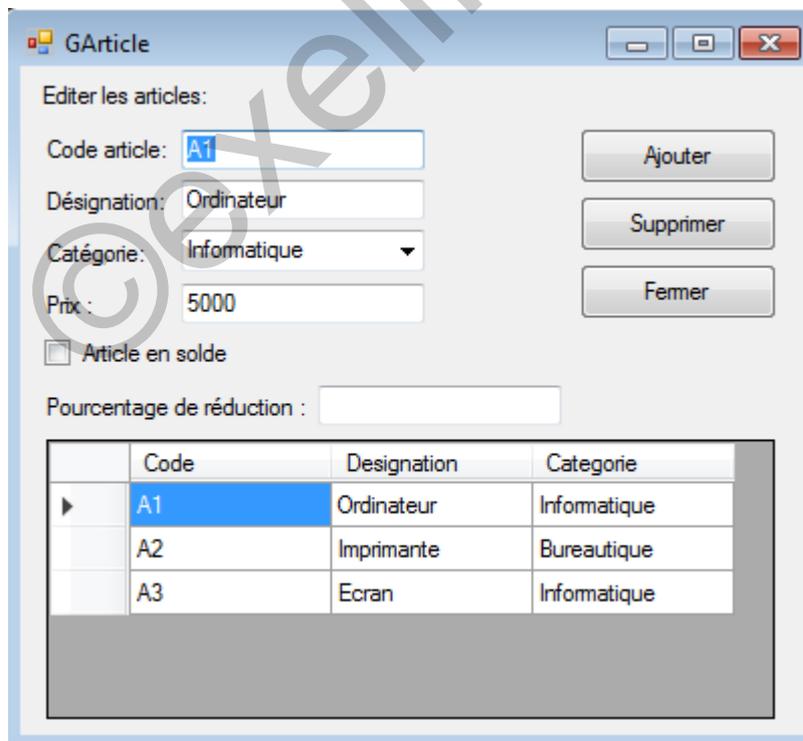
Intitulé du module : Programmation événementielle

Barème : / 40

On souhaite développer une application Windows pour la gestion des articles en stock d'un magasin en se basant sur la classe Article développée dans la partie P.O.O.

On dispose dans notre application d'un formulaire de gestion des articles. Le programme doit gérer une collection des articles disponibles dans le stock.

Le formulaire de gestion des articles est représenté ci-dessous :



The screenshot shows a Windows application window titled "GArticle". The window contains a form for editing articles with the following fields and controls:

- Code article:
- Désignation:
- Catégorie:
- Prix:
- Article en solde
- Pourcentage de réduction:

On the right side of the form, there are three buttons: "Ajouter", "Supprimer", and "Fermer".

Below the form is a table with the following data:

	Code	Designation	Categorie
▶	A1	Ordinateur	Informatique
	A2	Imprimante	Bureautique
	A3	Ecran	Informatique

Les contrôles du formulaire sont décrits dans le tableau suivant :

Type de contrôle	Nom du contrôle (Propriété name)	Contenu initial
TextBox	txt_code	Rien
TextBox	txt_designation	Rien
ComboBox	cb_categorie	Informatique, Bureautique
TextBox	txt_prix	0
CheckBox	cb_solde	Article en solde
TextBox	txt_remise	Rien
Button	btn_ajouter	Ajouter
Button	btn_supprimer	Supprimer
Button	btn_fermer	Fermer
DataGridView	dgv_articles	Rien

- 1) Déclarer dans la classe Program une collection d'articles en stock, cette collection doit être accessible partout dans le programme. **(2 Pts)**
- 2) Ecrire le code de la procédure événementielle qui s'exécute au chargement du formulaire permettant de :
 - a. Remplir la liste déroulante cb_categorie par les valeurs Informatique et Bureautique. **(2 Pts)**
 - b. Initialiser la zone de saisie txt_prix par 0. **(1 Pt)**
 - c. Désactiver la zone de saisie txt_remise. **(1 Pt)**
- 3) Ecrire le code de la procédure événementielle du bouton btn_ajouter permettant de :
 - a. Vérifier si tous les champs sont remplis. **(3 Pts)**
 - b. Vérifier si le prix entré par l'utilisateur est correcte. **(2 Pts)**
 - c. Vérifier si la remise entrée par l'utilisateur est correcte si le bouton cb_solde est coché. **(2 Pts)**
 - d. Vérifier si le code d'article entré par l'utilisateur n'existe pas dans la collection d'articles en stock. **(3 Pts)**
 - e. De créer un article avec les champs renseignés par l'utilisateur et l'ajouter à la collection d'articles en stock. **(3 Pts)**
 - f. De mettre à jour la grille dgv_articles. **(2 Pt)**
- 4) Ecrire le code de la procédure événementielle du bouton btn_supprimer permettant de :
 - a. Vérifier si le champ txt_code est rempli. **(1 Pt)**
 - b. Si le code d'article renseigné dans le champ txt_code correspond bien à un article dans la collection. **(2 Pts)**
 - c. D'afficher un message de confirmation « Voulez-vous vraiment supprimer l'article ? » à l'utilisateur. **(2 Pts)**
 - d. De supprimer l'article de la collection. **(2 Pts)**
 - e. De mettre à jour la grille dgv_articles. **(2 Pt)**
- 5) Ecrire le code de la procédure événementielle du bouton à cocher cb_solde permettant d'activer ou de désactiver le champ txt_remise selon l'état du bouton à cocher. **(2 Pts)**
- 6) Ecrire le code de la procédure événementielle au changement de la ligne sélectionnée dans la grille permettant de :
 - a. Remplir les champs par les valeurs des propriétés de l'article sélectionné. **(3 Pts)**
 - b. Tenir compte s'il s'agit d'un article en solde ou d'un article non soldé pour cocher ou décocher le bouton cb_solde et afficher la remise. **(2 Pts)**
- 7) Ecrire le code de la procédure événementielle du bouton btn_fermer permettant de quitter l'application après un message de confirmation. **(3 Pts)**