

Filière : Techniques de Développement Informatique

Niveau: TS

Durée : 5 heures

Epreuve : Synthèse

Variante : V2

Barème : / 120Pts

❖ Partie I : Théorie (40 pts)

➤ Dossier 1: Notions de mathématiques appliquées à l'informatique (12 pts)

Exercice 1:(6 pts)

NB : la calculatrice est strictement interdite.

Compléter le tableau suivant:

Décimal	Binaire	Octal	Hexadécimal
2379			
	11011011		
		1534	
			13E

Exercice 2: Soit la fonction logique suivante:

$$F(A, B, C) = AC + (\bar{A} + \bar{C})B + \overline{A + \bar{B} + \bar{C}}$$

- 1- Simplifier analytiquement la fonction logique **Q. (2 pts)**
- 2- Construire la table de vérité. **(2 pts)**
- 3- Simplifier avec la méthode de Karnaugh la fonction logique **Q. (2 pts)**

➤ Dossier 2: Techniques de programmation structurée (16 pts)

Exercice 1:

Quel résultat fournit l'algorithme suivant **(6 pts)**

Filière	Epreuve	Session	1/6
DI	Synthèse V2	Juillet 2015	

Variables i, j : entier
ch : chaîne de caractères

Début

```

pour i = 1 à 8 faire
    ch ← ""
    pour j = 1 à 8 faire
        si ((i + j) % 2 = 0) alors
            ch ← ch & "#"
        sinon
            ch ← ch & "O"
        finsi
    fin pour
    écrire(ch)
fin pour
fin

```

NB :

- & : permet la concaténation de chaînes.
- Le symbole % permet de calculer le reste de la division.
- **Ecrire(c)** : affiche la chaîne c et avec un retour à la ligne

Exercice 2: (10 pts)

Ecrire une procédure **SupSignes()** qui permet d'éliminer les signes de ponctuations (, ; . : ! ?) à partir d'un tableau de caractères. (6 pts)

Faites appel à cette fonction dans un algorithme donnant le droit à l'utilisateur de remplir un tableau avec N valeurs de type chaîne de caractères. (4 pts)

Exemple :

Le tableau saisi :

:

s	l	t	,	c	a	v	a	?
---	---	---	---	---	---	---	---	---

Tableau résultat :

s	l	t	c	a	v	a
---	---	---	---	---	---	---

Dossier 3: Conception et modélisation d'un système d'information (12 pts)

Conception d'un système d'information pour la gestion d'estivage

Une entreprise désire réaliser un intranet pour gérer les séjours de ses employés aux centres d'estivage. Chaque site se caractérise par un nom, se situe dans une ville donnée et

Filière	Epreuve	Session	2 / 6
DI	Synthèse V2	Juillet 2015	

contient plusieurs types de logement (appartement, studio, villa...) certains types bénéficient d'une cuisine équipée. Le prix de la nuitée dépend du type et du site.

Un employé qui désire bénéficier d'un séjour au sein d'un site d'estivage fait une demande. Vu le succès de certains types de séjours, il est conseillé de remplir trois choix par demande mais ce n'est pas une obligation. La demande précise également la période demandée ainsi que les personnes accompagnées. Pour ces dernières on doit enregistrer le nom, le prénom l'âge et leur degrés de parenté avec l'employé demandeur.

A Chaque moment on doit connaître l'état de la demande(en cours, accordée ou refusée). En cas de refus, on doit connaître le motif.

En ce qui concerne les employés, on veut mémoriser leur matricule, nom, prénom, fonction, situation familiale, adresse, n° de téléphone, ainsi que leur date de recrutement.

- a) Etablir le dictionnaire de données. (3 pts)
- b) Etablir le modèle conceptuel de données. (6 pts)
- c) Etablir le modèle logique de données. (3 pts)

❖ Partie II: Pratique (80 pts)

➤ Dossier 1: Langage de programmation structurée (20 pts)

On souhaite écrire un programme permettant de gérer les quartiers dans une commune. Pour cela nous allons considérer la structure **Quartier** ayant pour données :

- Une chaîne **Nom** qui correspond au nom du quartier
- Un entier **surface**, qui correspond à la surface du quartier associé à la structure,
- **Nbr_habitant** qui correspond au nombre d'habitants,

Pour la suite de l'exercice, on suppose que les **Quartiers** sont stockés dans un tableau **Les_Quartiers** de dimension maximale 50 et dont la taille réelle est enregistrée dans la variable n.

Ecrire un programme complet permettant la gestion des quartiers de la commune. Ce programme doit afficher au départ le menu suivant : (2 pts)

<i>Filière</i>	<i>Epreuve</i>	<i>Session</i>	3 / 6
<i>DI</i>	<i>Synthèse V2</i>	<i>Juillet 2015</i>	


```

*****Gestion des quartiers *****
Taper le numero de l'operation a realiser :
1 ----- Ajouter un nouveau quartier. (4 pts)
2 ----- Afficher les quartiers. (3 pts)
3 ----- Supprimer un quartier. (3 pts)
4 ----- Chercher un quartier. (3 pts)
5 ----- Sauvegarder dans un fichier. (4 pts)
6 ----- Quitter (1 pt)
*****

```

Ci-dessous l'explication du menu :

1. Ajouter un nouveau quartier au tableau des quartiers dont les informations sont saisies au clavier.
2. Afficher tous les quartiers.
3. Supprimer le quartier dont le nom est donné par l'utilisateur.
4. Afficher les quartiers dont le nombre d'habitants est inférieur à une valeur saisie.
5. Copier le contenu de tableau dans un fichier texte dont le nom est saisi par l'utilisateur. Un quartier par ligne, les champs sont séparés par le caractère « \ ».

➤ Dossier 2: Programmation orientée objet (30 pts)

Développement d'une application orientée objet pour la gestion des appareils électriques

On désire réaliser une application pour une société qui fabrique et commercialise des appareils électriques :

Classe Appareil : On considère qu'un appareil est caractérisé par une référence, une puissance (exprimée en watt), un poids et un prix.

1)

- a. Ecrire la classe « **AppElectrique** » permettant de modéliser ces objets. (2 pts)

Ajouter à cette classe un constructeur permettant d'instancier des objets de la classe « **AppElectrique** » dont on précisera la référence, la puissance, le poids, et un constructeur sans paramètres. (2 pts).

- b. Réécrire la méthode **ToString()** pour afficher les caractéristiques de l'appareil. (2 pts)
- c. Ecrire la méthode **ClasseEnergetique()** : qui permet d'afficher la classe énergétique de l'appareil selon sa puissance : (2 pts)
- Si la puissance est inférieure strictement à 300 watt c'est la classe A

Filière	Epreuve	Session	4 / 6
DI	Synthèse V2	Juillet 2015	

- Si la puissance est comprise entre 300 watt et 1000 watt c'est la classe B
- Si la puissance est supérieure à 1000 watt c'est la classe C

- 2) **Classe Television** : une télévision est un appareil qui possède un type d'écran (LCD, LED ...), et une fréquence(en hertz) :
 - a) Ecrire une classe « **Television** » héritant de la classe « **AppElectrique** ». (2 pts)
 - b) Récrire le constructeur de cette classe afin d'initialiser, en plus, le type d'écran et la fréquence avec des valeurs passées en paramètre. (2 pts)
 - c) Réécrire la méthode **toString()** affichant les caractéristiques de la télévision. (2 pts)
- 3) **Classe VeloElec** : un vélo électrique est un appareil qui a une autonomie et un kilométrage.
 - a) Ecrire une classe **VeloElec** héritant de la classe « **AppElectrique** ». (2 pts)
 - b) Récrire le constructeur de cette classe pour définir, en plus, l'autonomie (exprimée en Km) et le kilométrage. (2 pts)
 - c) Ajouter les méthodes suivantes :
 - i) **Rouler (float distance)** : permettant au vélo d'avancer et retournant le nouveau kilométrage. (2 pts)
 - ii) **Charger (int nbrminute)** : permettant de charger la batterie et renvoyer la nouvelle valeur de l'autonomie (on suppose qu'une heure de charge donne 10 km en autonomie). (2 pts)
 - d) Réécrire la méthode **toString()** donnant les caractéristiques d'un vélo électrique. (2 pts)
- 4) **Classe Program** : Tester ces trois classes dans un programme principal.
 - a) Créer un appareil électrique de type télévision (1 pt)
 - b) Afficher cet article et afficher sa classe énergétique. (1,5 pts)
 - c) Créer un article de type vélo électrique (1 pt)
 - d) Faites rouler ce vélo (1 pt)
 - e) Charger ce vélo (1 pt)
 - f) Afficher le vélo (0,5 pt)

➤ Dossier 3: Programmation événementielle (30 pts)

- **NB** : Dans ce dossier, on vous demande de donner uniquement le code à mettre à l'intérieur des méthodes événementielles. L'entête de ces méthodes événementielles n'est pas demandé !

Soit le formulaire suivant dont l'objectif consiste à gérer des logements :

<i>Filière</i>	<i>Epreuve</i>	<i>Session</i>	5 / 6
<i>DI</i>	<i>Synthèse V2</i>	<i>Juillet 2015</i>	

liste des logements

ID BATIMENT: AP2 Nouveau

ADRESSE: 12 hay amal casa Ajouter

SUPERFICIE: 120 Supprimer

TYPE: Appartement ▼ Sauvegarder

PRIX: 150000000 Dhs

	id_batiment	adresse	superficie	type	prix
	S1	12 MAARIF CASA	50	Studio	100000000
	AP1	34 RIYAD RABAT	200	Appartement	350000000
>*					

1. Ajouter dans la méthode de chargement du formulaire le code permettant de remplir la liste type par (Studio, Appartement, Maison). (3 pts)
2. Ecrire le code du bouton « **Nouveau** » qui permet d'initialiser tous les champs pour saisir un nouveau logement. (3 pts)
3. Ecrire le code du bouton « **Ajouter** » permettant d'ajouter un nouveau logement à la DataGridView des logements. (8 pts)
4.
 - a. Ecrire le code du bouton « **Supprimer** » permettant de supprimer de la DataGridView un logement dont l'ID est saisi. (6 pts)
 - b. Un message demandant la confirmation doit être affiché. (2 pts)
5. On suppose que les logements sont stockées dans une collection «Logements» de type ArrayList ou Vector.
Ecrire le code du bouton « **Sauvegarder** » qui permet d'enregistrer la liste des logements (voir image ci-dessus) dans un fichier *objet* (sérialisation) qui porte le nom « **Logements.dat** ». (8 pts)

Filière	Epreuve	Session	6/6
DI	Synthèse V2	Juillet 2015	