



مكتب التكوين المهني وإنعاش الشغل

Office de la Formation Professionnelle
et de la Promotion du Travail

Direction Recherche et Ingénierie de la Formation

Examen Passage
Session Juin 2011

Filière : Techniques de Développement Informatique

Epreuve : Pratique – V1-1

Barème : 60 points

Niveau : Technicien Spécialisé

Durée : 4h30mn

Variante 1-1

Important : assurez-vous que tous les éléments de vos projets sont sauvegardés dans un dossier portant votre nom et prénom et le numéro de la variante de l'examen comme suit "NOM PRENOM Variante".

Partie 1 : Programmation Structurée (15 pts)

1. Ecrire un programme en C ou équivalent permettant de compter le nombre de diviseurs d'un nombre entier donné par l'utilisateur, puis de les afficher du plus grand au plus petit. (5 pts)

Exemple d'exécution :

Le nombre de diviseurs de 8 est : 4
Les diviseurs de 8 par ordre décroissant : 8, 4, 2, 1

2. Ecrire un programme en C ou équivalent permettant de saisir un tableau de 10 valeurs entières, et de diviser l'ensemble des éléments du tableau par la valeur du n^{ème} élément du tableau, n étant saisi par l'utilisateur. (4 pts)

Exemple d'exécution :

Soit le tableau saisi : 8 - 4 - 6 - 2 - 4 - 12 - 6 - 14 - 10 - 8
n = 4 :
Tableau résultat : 4 - 2 - 3 - 1 - 2 - 6 - 3 - 7 - 5 - 4

- 3.
- a- Ecrire une fonction nommée « puissance(x,y) » permettant de retourner x^y , avec x et y des réels passés en paramètres . (2 pts)
- b- Ecrire un programme qui utilise la fonction « puissance» de la question a) pour calculer l'équivalent décimal d'un nombre binaire. (4 pts)

Rappel : $(101101)_2 = 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 + 0 \times 2^4 + 1 \times 2^5 = (45)_{10}$

Exemple d'exécution :

Entrez le nombre binaire : 101101 ↵
L'équivalent décimal est : 45

Partie 2 : Programmation Orientée Objet (25 pts)

L'objet de cette application est la gestion informatisée du cabinet médical. Elle doit procéder à la gestion informatisée des dossiers médicaux des patients et elle s'occupe également de la gestion des rendez-vous et des visites.

1. Classe Patient. (2,5 pts)

- a. Définir une classe Patient dont les caractéristiques sont : Code patient (affecté de façon incrémentale par rapport au nombre de patients), Nom, Prénom, Date de naissance, Adresse, Tél, E-mail. (0.5 pt)
- b. Écrire un constructeur à deux paramètres : Nom et prénom. (0.5 pt)
- c. Écrire un deuxième constructeur avec tous les paramètres. (0.5 pt)
- d. Écrire les accesseurs des champs et la méthode **toString()** qui renverra les informations d'un patient . (1 pt)

2. Classe Visites. (2,5 pts)

- a. Définir une classe Visites dont les caractéristiques sont Date visite, Heure visite, Code patient, Montant payé. (0.5 pt)
- b. Écrire un constructeur à deux paramètres : Date visite et Heure visite. (0.5 pt)
- c. Écrire un deuxième constructeur sans paramètres qui appelle le premier avec la date et l'heure actuelle. (0.5 pt)
- d. Écrire les accesseurs des champs et la méthode **toString()** qui renverra les informations de cette visite . (1 pt)

3. Classe RendezVous (2 pts)

- a. Définir une classe RendezVous dont les caractéristiques sont : Date RendezVous, Heure RendezVous , Code patient, Observation. (0.5 pt)
- b. Écrire un constructeur à trois paramètres : Date RendezVous , Heure RendezVous et code de patient. (0.5 pt)
- c. Écrire les accesseurs des champs et la méthode **toString()** qui renverra les informations de ce rendez-vous. (1 pt)

4. Classe CabinetMedical (18 pts)

- a. Définir une classe **CabinetMedical** dont les caractéristiques sont : une liste des patients, une liste des visites et une liste des rendez-vous (collection ou tableaux). (1 pt)
- b. Ajouter une méthode **ajouterPatient** qui ajoute un patient à l'ensemble des patients du cabinet. (2 pts)
- c. Ajouter une méthode **patientExitant** ayant comme paramètre le nom et le prénom et qui retourne le code du patient s'il existe et le chiffre « -1 » sinon. (2 pts)
- d. Définir une classe d'exception : **exceptionMedecinOccupe**. (1 pt)
- e. Ajouter une méthode **ajouterRDV** qui ajoute un rendez-vous, et lève l'exception définie en d) s'il coïncide avec un autre rendez-vous. (2 pts)
- f. Ajouter une méthode **afficherRDVDuJour** qui va, pour une date donnée comme paramètre, afficher la liste des rendez-vous de ce jour. (2 pts)
- g. Ajouter une méthode **patientAyantDesVisites** qui affiche la liste des patients (code, nom et prénom) ayant visité le cabinet pendant la dernière semaine. (3 pts)
- h. Ajouter une méthode **supprimerPatient** qui permet de supprimer un patient ainsi que ses visites et rendez-vous. (2 pts)
- i. Ajouter une méthode **enregistrerPatients** qui permet d'enregistrer la liste des patients dans un fichier. (3 pts)

Partie 3 : Programmation Evénementielle (20 pts)

Soit l'interface d'inscription suivante :

Inscription

Nom :	<input type="text"/>			
Prénom :	<input type="text"/>			
Sexe:	<input type="radio"/> M	<input checked="" type="radio"/> F		
Option :	<input style="width: 100px; height: 20px; border: none; background-color: #f0f0f0; padding: 2px 10px; border-radius: 5px; font-size: 10px; font-weight: bold; margin-right: 10px;" type="button" value="TDI"/>			
Age :	<input style="width: 40px; height: 20px; border: none; background-color: #f0f0f0; padding: 2px 10px; border-radius: 5px; font-size: 10px; font-weight: bold; margin-right: 10px;" type="button" value="0"/>	<input style="width: 100px; height: 20px; border: none; background-color: #f0f0f0; padding: 2px 10px; border-radius: 5px; font-size: 10px; font-weight: bold; margin-right: 10px;" type="button" value="Ajouter"/>		
<input style="width: 100px; height: 20px; border: none; background-color: #f0f0f0; padding: 2px 10px; border-radius: 5px; font-size: 10px; font-weight: bold; margin-right: 10px;" type="button" value="Supprimer"/>				
Nom	Prénom	Sexe	Option	Age
<input type="text"/>	<input type="text"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>

- 1- Créer cette interface. (4 pts)
- 2- Créer une classe Stagiaire caractérisée par : Nom, prénom, sexe, option et âge. (2 pts)
- 3- Ecrire le code correspondant au bouton **Ajouter** qui permet d'ajouter un stagiaire dans une collection d'objets Stagiaires et de l'afficher dans la grille. (3 pts)
- 4- Ecrire le code correspondant au bouton **Supprimer** qui permet de supprimer un stagiaire sélectionné dans la grille, l'application doit afficher un message de confirmation avant de supprimer le stagiaire et un message d'erreur si aucune ligne n'est sélectionnée. (4 pts)
- 5- Ajouter un bouton **Rechercher** qui permet de rechercher un stagiaire par nom saisi dans la zone de texte « Nom » et d'afficher toutes les informations concernant ce stagiaire au niveau des zones correspondantes de l'interface. (4 pts)
- 6- Créer un bouton **Afficher** qui permet d'afficher dans la grille la liste des stagiaires d'une option donnée choisie dans la liste déroulante « Option ». (3 pts)



مكتب التكوين المهني وإنعاش الشغل

Office de la Formation Professionnelle
et de la Promotion du Travail

Direction Recherche et Ingénierie de la Formation

Examen Passage
Session Juin 2011

Filière : Techniques de Développement Informatiques

Epreuve : Pratique – V1-2

Niveau : Technicien Spécialisé

Barème : 60 points

Durée : 4h30mn

Variante 1-2

Important : assurez-vous que tous les éléments de vos projets sont sauvegardés dans un dossier portant votre nom et prénom et le numéro de la variante de l'examen comme suit "NOM PRENOM Variante".

Partie 1 : Programmation Structurée (15 pts)

1. Ecrire un programme en C ou équivalent permettant de compter le nombre de multiples (inférieurs à 1000) d'un nombre entier donné par l'utilisateur, puis de les afficher du plus grand au plus petit. (5 pts)

Exemple d'exécution :

Le nombre de multiples (< 1000) de 120 est : 8
Les multiples de 120 par ordre décroissant :
960 - 840 - 720 - 600 - 480 - 360 - 240 - 120

2. Ecrire un programme en C ou équivalent permettant de saisir un tableau de 10 valeurs entières, et de multiplier l'ensemble des éléments du tableau par la valeur du n^{ème} élément du tableau, n étant saisi par l'utilisateur. (4 pts)

Exemple d'exécution :

Soit le tableau saisi : 8 - 5 - 2 - 6 - 4 - 12 - 6 - 14 - 10 - 8
n = 3 :
Tableau résultat : 16 - 10 - 4 - 12 - 8 - 24 - 12 - 28 - 20 - 16

- 3.
- a- Ecrire une fonction nommée « puissance(x,y) » permettant de retourner x^y , avec x et y des réels passés en paramètres . (2 pts)
- b- Ecrire un programme qui utilise la fonction « puissance» de la question a) pour calculer l'équivalent décimal d'un nombre octal . (4 pts)

Rappel : $(177)_8 = 7 \times 8^0 + 7 \times 8^1 + 1 \times 8^2 = (127)_{10}$

Exemple d'exécution :

Entrez le nombre octal : 177 ↵
L'équivalent décimal est : 127

Partie 2 : Programmation Orientée Objet (25 pts)

L'objet de cette application est la gestion informatisée du cabinet d'avocat. Elle doit procéder à la gestion informatisée des dossiers des clients et elle s'occupe également de la gestion des rendez-vous et des visites.

1. Classe Client. (2,5pts)

- a. Définir une classe Client dont les caractéristiques sont : Code client (affecté de façon incrémentale par rapport au nombre de clients), Nom, Prénom, Date de naissance, Adresse, Tél, E-mail. (0,5 pt)
- b. Ecrire un constructeur à deux paramètres : Nom et prénom. (0,5 pt)
- c. Écrire un deuxième constructeur avec tous les paramètres. (0,5 pt)
- d. Écrire les accesseurs des champs et la méthode **toString** qui renverra les informations d'un client. (1 pt)

2. Classe Visites. (2,5pts)

- a. Définir une classe Visites dont les caractéristiques sont Date visite, Heure visite, Code client, sujet de visite. (0,5 pt)
- b. Ecrire un constructeur à deux paramètres : Date de visite et Heure visite. (0,5 pt)
- c. Écrire un deuxième constructeur sans paramètres qui appelle le premier avec la date et l'heure actuelle. (0,5 pt)
- d. Écrire les accesseurs des champs et la méthode **toString** qui renverra les informations de cette visite. (1 pt)

3. Classe Dossier(2pts)

- a. Définir une classe Dossier dont les caractéristiques sont : Code client, tribunal, sujet, jugement, et statut (en cours ou clôturé). (0,5 pt)
- b. Ecrire un constructeur à deux paramètres : Code client et sujet. (0,5 pt)
- c. Écrire les accesseurs des champs et la méthode **toString** qui renverra les informations d'un dossier. (1 pt)

4. Classe CabinetAvocat (18pts)

- a. Définir une classe **CabinetAvocat** dont les caractéristiques sont : une liste des clients, une liste des visites et une liste des dossiers (collection ou tableaux). (1 pt)
- b. Ajouter une méthode **ajouterClient** qui ajoute un client à l'ensemble des clients du cabinet. (2 pts)
- c. Ajouter une méthode **clientExitant** ayant comme paramètre le nom et le prénom et qui retourne « vrai » si l'existe et « faux » sinon. (2 pts)
- d. Définir une classe d'exception : **exceptionClient**. (1 pt)
- e. Ajouter une méthode **ajouterVisite** qui ajoute une visite, et lève l'exception définie en d) si le code du client n'existe pas. (2 pts)
- f. Ajouter une méthode **afficherDossier** qui affiche la liste des dossiers d'un client donné. (2 pts)
- g. Ajouter une méthode **clientAyantDesVisites** qui affiche la liste des clients (code, nom et prénom) ayant visité le cabinet pendant la dernière semaine. (3 pts)
- h. Ajouter une méthode **supprimerDossiers** qui permet de supprimer les dossiers clôturés. (2 pts)
- i. Ajouter une méthode **enregistrer** qui permet d'enregistrer la liste des clients, liste des dossiers dans un fichier. (3 pts)

Partie 3 : Programmation Evénementielle (20 pts)

Soit l'interface d'inscription suivante :

Matricule	Marque	Energie	Couleur	Puissance

- 1- Créer cette interface. (4 pts)
- 2- Créer une classe Voiture caractérisée par : Matricule, Marque, Carburant (Essence ou Diesel), Couleur et Puissance. (2 pts)
- 3- Ecrire le code correspondant au bouton **Ajouter** qui permet d'ajouter une voiture dans une collection d'objets Voitures et de l'afficher dans la grille. (3 pts)
- 4- Ecrire le code correspondant au bouton **Supprimer** qui permet de supprimer une voiture sélectionnée dans la grille, l'application doit afficher un message de confirmation avant de supprimer la voiture et un message d'erreur si aucune ligne n'est sélectionnée. (4 pts)
- 5- Ajouter un bouton **Rechercher** qui permet de rechercher une voiture par Matricule saisi dans la zone de texte « Matricule » et d'afficher toutes les informations concernant cette voiture au niveau des zones correspondantes de l'interface. (4 pts)
- 6- Créer un bouton **Afficher** qui permet d'afficher dans la grille la liste des voitures d'une marque donnée saisie dans la zone « Marque ». (3 pts)

Barème de notation :

Partie 1 : (15 pts)

Question	Note
1	5
2	4
3	6

Partie 2 : (25 pts)

Question	Note
1. a	0.5
1. b	0.5
1. c	0.5
1. d	1
2. a	0.5
2. b	0.5
2. c	0.5
2. d	1
3. a	0.5
3. b	0.5
3. c	1
4. a	1
4. b	2
4. c	2
4. d	1
4. e	2
4. f	2
4. g	3
4. h	2
4. i	3

Partie 3 : (20 pts)

Question	Note
1	4
2	2
3	3
4	4
5	4
6	3



مكتب التكوين المهني وإنعاش الشغل

Office de la Formation Professionnelle
et de la Promotion du Travail

Direction Recherche et Ingénierie de la Formation

Examen Passage
Session Juin 2011

Filière : Techniques de Développement Informatiques

Epreuve : Pratique – V1-3

Barème : 60 points

Niveau : Technicien Spécialisé

Durée : 4h30mn

Variante 1-3

Important : assurez-vous que tous les éléments de vos projets sont sauvegardés dans un dossier portant votre nom et prénom et le numéro de la variante de l'examen comme suit "NOM PRENOM Variante".

Partie 1 : Programmation Structurée (15 pts)

1. Ecrire un programme en C ou équivalent permettant de compter le nombre de puissances successives (inférieurs à 1000) d'un nombre entier donné par l'utilisateur, puis de les afficher du plus grand au plus petit. (5 pts)

Rappel : Les puissances successives de 2 sont : $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, 8 , 16 , 32 , 64 , 128 ...

Exemple d'exécution :

Le nombre de puissances successives (<1000) de 5 est : 5
Les puissances successives de 5 par ordre décroissant : 625, 125, 25, 5, 1

2. Ecrire un programme en C ou équivalent permettant de saisir un tableau de 10 valeurs entières, et d'ajouter à l'ensemble des éléments du tableau la valeur du n^{ème} élément du tableau, n étant saisi par l'utilisateur. (4 pts)

Exemple d'exécution :

Soit le tableau saisi : 8 - 4 - 6 - 3 - 7 - 12 - 6 - 14 - 10 - 8
n = 5 :
Tableau résultat : 15 - 11 - 13 - 10 - 14 - 19 - 13 - 21 - 17 - 15

3.

- a- Ecrire une fonction nommée « puissance(x,y) » permettant de retourner x^y , avec x et y des réels passés en paramètres . (2 pts)

- b- Ecrire un programme qui utilise la fonction « puissance» de la question a) pour calculer la somme : $1/x^0 + 1/x^1 + 1/x^2 + \dots + 1/x^n$ pour x réel donné et n entier donné saisis au clavier. (4 pts)

Exemple d'exécution :

Entrez la valeur de x : 3 ↵
Entrez la valeur de n : 5 ↵
La somme est : 1.497942

Partie 2 : Programmation Orientée Objet (25 pts)

L'objet de cette application est la gestion informatisée du cabinet médical. Elle doit procéder à la gestion informatisée des dossiers médicaux des patients et elle s'occupe également de la gestion des rendez-vous et des visites.

1. Classe Patient. (2,5pts)

- a. Définir une classe **Patient** dont les caractéristiques sont : Code patient (affecté de façon incrémentale par rapport au nombre de patients), Nom, Prénom, Date de naissance, Adresse, Tél, E-mail. (0,5 pt)
- b. Ecrire un constructeur à deux paramètres : Nom et prénom. (0,5 pt)
- c. Écrire un deuxième constructeur avec tous les paramètres. (0,5 pt)
- d. Écrire les accesseurs des champs et la méthode **afficher** qui affiche les informations d'un patient. (1 pt)

2. Classe Visites. (2,5pts)

- a. Définir une classe **Visites** dont les caractéristiques sont Date visite, Heure visite, Code patient, Montant payé. (0,5 pt)
- b. Ecrire un constructeur à trois paramètres : Date visite et Heure visite et code patient. (0,5 pt)
- c. Écrire un deuxième constructeur à un seul paramètre : code client et qui initialise la date et l'heure avec la date et l'heure actuelle. (0,5 pt)
- d. Écrire les accesseurs des champs et la méthode **afficher** qui affiche les informations de cette visite. (1 pt)

3. Classe RendezVous(2pts)

- a. Définir une classe **RendezVous** dont les caractéristiques sont : Date RendezVous, Heure RendezVous , Code patient, Observation. (0,5 pt)
- b. Ecrire un constructeur à trois paramètres : Date RendezVous ,Heure RendezVous et code de patient. (0,5 pt)
- c. Écrire les accesseurs des champs et la méthode **afficher** qui affiche les informations de ce rendez-vous. (1 pt)

4. Classe CabinetMedical (18pts)

- a. Définir une classe **CabinetMedical** dont les caractéristiques sont : une liste des patients, une liste des visites et une liste des rendez-vous. (1 pt)
- b. Ajouter une méthode **ajouterPatient** qui ajoute un patient à l'ensemble des patients du cabinet. (2 pts)
- c. Ajouter une méthode **patientExitant** ayant comme paramètre le nom et le prénom et qui retourne « vrai » si le patient existe et « faux » sinon. (2 pts)
- d. Définir une classe d'exception : **exceptionMontant**. (1 pt)
- e. Ajouter une méthode **ajouterVisite** qui ajoute une visite, et lève l'exception définie en d) si le montant payé est négatif. (2 pts)
- f. Ajouter une méthode **recetteDuJour** qui, pour une date donnée comme paramètre, affiche la somme d'argent encaissée ce jour. (2 pts)
- g. Ajouter une méthode **patientAyantDesVisites** qui affiche la liste des patients ayant visité le cabinet pendant la dernière semaine. (3 pts)
- h. Ajouter une méthode **annulerRDV** qui permet d'annuler un rendez-vous. (2 pts)
- i. Ajouter une méthode **exporterRDV** qui permet d'enregistrer la liste des rendez-vous dans un fichier. (3 pts)

Partie 3 : Programmation Evénementielle (20 pts)

Soit l'interface d'inscription suivante :

Inscription

Nom :	<input type="text"/>																														
Prénom :	<input type="text"/>																														
Sexe:	<input type="text" value="Homme"/>																														
Option :	<input checked="" type="radio"/> TDI <input type="radio"/> TRI <input type="radio"/> TDM																														
Age :	<input type="text" value="0"/>																														
<input type="button" value="Ajouter"/> <input type="button" value="Supprimer"/>																															
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Nom</th> <th style="text-align: left; padding: 2px;">Prénom</th> <th style="text-align: left; padding: 2px;">Sexe</th> <th style="text-align: left; padding: 2px;">Option</th> <th style="text-align: left; padding: 2px;">Age</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table>		Nom	Prénom	Sexe	Option	Age																									
Nom	Prénom	Sexe	Option	Age																											

- 1- Créer cette interface. (4 pts)
- 2- Créer une classe Stagiaire caractérisée par : Nom, prénom, sexe, option et âge. (2 pts)
- 3- Ecrire le code correspondant au bouton **Ajouter** qui permet d'ajouter un stagiaire dans une collection d'objets Stagiaires et de l'afficher dans la grille. (3 pts)
- 4- Ecrire le code correspondant au bouton **Supprimer** qui permet de supprimer un stagiaire sélectionné dans la grille, l'application doit afficher un message de confirmation avant de supprimer le stagiaire et un message d'erreur si aucune ligne n'est sélectionnée. (4 pts)
- 5- Ajouter un bouton **Rechercher** qui permet de rechercher un stagiaire par nom saisi dans la zone de texte « Nom » et d'afficher toutes les informations concernant ce stagiaire au niveau des zones correspondantes de l'interface. (4 pts)
- 6- Créer un bouton **Afficher** qui permet d'afficher dans la grille la liste des stagiaires d'une option donnée choisie parmi les boutons d'options. (3 pts)

Barème de notation :

Partie 1 : (15 pts)

Question	Note
1	5
2	4
3	6

Partie 2 : (25 pts)

Question	Note
1. a	0.5
1. b	0.5
1. c	0.5
1. d	1
2. a	0.5
2. b	0.5
2. c	0.5
2. d	1
3. a	0.5
3. b	0.5
3. c	1
4. a	1
4. b	2
4. c	2
4. d	1
4. e	2
4. f	2
4. g	3
4. h	2
4. i	3

Partie 3 : (20 pts)

Question	Note
1	4
2	2
3	3
4	4
5	4
6	3