

CONCOURS d'ACCES AUX ECOLES SUPERIEURES EN INFORMATIQUE

Epreuve : Informatique

Code : Info

Date : 20/06/2017

Durée : 2h30

Coefficient : 4

Instructions Générales (à lire avant le début de l'épreuve)

- Les candidats doivent vérifier que le sujet comprend 6 pages.
- Les candidats sont invités à porter une attention particulière à la présentation.
- Les candidats doivent rendre les copies même vierges.
- Si au cours de l'épreuve, un candidat repère ce qui lui semble être une erreur d'énoncé, il le signalera sur sa copie et devra poursuivre sa composition en expliquant les raisons des initiatives qu'il a été amené à prendre.
- Les numéros des questions doivent être transcrits clairement sur les copies.
- Les pages des copies et des feuilles intermédiaires doivent être numérotées (1, 2, 3, 4,...).
- Les documents sont interdits, sauf indication contraire sur le sujet.
- L'emploi d'une calculatrice n'est pas autorisé.
- Aucun échange n'est autorisé entre les candidats.
- Les parties I et II sont indépendantes et le candidat peut les traiter dans l'ordre de son choix.
- Les parties I et II doivent être rédigées sur des copies séparées.

Barème de notation :

- Partie I : 12 points

- Partie II : 8 points,

Partie I : Algorithmique et Programmation (12 points).

Exercice 1 :

On voudrait représenter les dairas des 48 wilayas dans des structures de données statiques.

La wilaya étant composée de dairas.

Nous allons utiliser des structures de données statiques.

Chaque wilaya sera représentée par la structure suivante :

Nom de la Wilaya, le nombre de dairas et l'ensemble des noms de Dairas rattachées à cette wilaya.

1. Définir la structure de données statiques pour représenter les 48 Wilayas. (Le nombre maximum de dairas par wilaya est de 10.) (1 pt)
2. Donner l'algorithme de création des wilayas et de leur dairas. (1.5 pts)
3. Comment retrouver la wilaya d'une daïra donnée ? (1.5 pts)

Exercice 2 :

L'insertion dans un arbre binaire de recherche est-elle commutative ? Autrement dit, a-t-on le même arbre si on insère deux éléments X1 ensuite Y1 ou Y1 ensuite X1 dans cet arbre? (0.5 pt)

Soit un arbre binaire de recherche R d'entiers.

Ecrire une procédure qui affiche tous les éléments de cet arbre supérieurs à la valeur V. Le parcours doit être réduit au maximum. (1.5 pts)

Ecrire une fonction récursive qui affiche les valeurs de cet arbre R dans l'ordre croissant. (1pt)

Ecrire une fonction qui vérifie que l'ordre infixe d'un arbre binaire de recherche est l'ordre croissant sur ses valeurs. (1pt)

Exercice 3 :

Nous souhaitons développer un système de gestion de stagiaires pour une formation payante. Tous les stagiaires ont un identifiant ID, et une note qui sera attribuée par le formateur. Pour certains inscrits (les boursiers) une note minimale à atteindre est fixée lors de leur inscription. Si la note obtenue est inférieure à la note minimale, un traitement est déclenché.

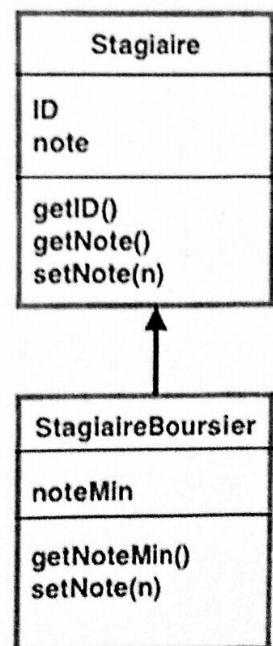
La figure suivante donne une partie du diagramme des classes:

1. a. Pourquoi a-t-on redéfini la méthode setNote(n) dans la classe StagiaireBoursier?

b. Quel principe de la POO a-t-on appliqué ?

Lorsque la note d'un stagiaire boursier est inférieure à la note minimale attendue, le système lance une exception NoteInférieureException à travers laquelle un rendez-vous est pris avec un membre de l'équipe pédagogique: le tuteur, le formateur ou le directeur des études, selon le rapport entre la note obtenue et la note minimale :

- le tuteur se charge des cas où ce rapport (note obtenue/note minimale) est supérieur à 2/3 ;
- le formateur prend en charge ceux pour lesquels ce rapport est supérieur à 1/3 ;
- le directeur des études prend en charge les autres cas.



Pour gérer les rendez-vous des membres de l'équipe pédagogique, nous utilisons :

- la classe RDV dont le constructeur crée un rendez-vous pour un étudiant donné
- la classe EquipePdagogique dotée d'attributs de type ArrayList permettant la sauvegarde des rendez-vous de chacun des trois membres de l'équipe, et de la méthode statique `prendreRDV(Etudiant e, double rapport)` qui crée un rendez-vous pour l'étudiant *e* et l'ajoute à la liste des rendez-vous de l'un des trois membres précités. (0.5 pt)

2. Donnez les portions du programme nécessaires pour la gestion des rendez-vous en donnant le code java de: (3.5 pts)

- a. La classe `NoteInferieurException`
- b. La méthode `setNote(double n)` de la classe `StagiaireBoursier`,
- c. Les attributs de la classe `EquipePedagogique`,
- d. La méthode `prendreRDV(Etudiant e, double rapport)`

Partie II : Architecture des ordinateurs (8 points).

Exercice 1: (2 points)

On veut réaliser un circuit qui permet d'effectuer la division de deux nombres A et B de deux bits chacun : A (A1, A0) et B (B1, B0).

- Le circuit réalise : $(A1, A0) / (B1, B0)$
- Le résultat de chaque opération est codé sur quatre bits chacun.

Question :

Réaliser la table de vérité de ce circuit, puis donner le schéma du circuit en utilisant une seule ROM de taille minimum et un minimum de circuits.

Exercice 2: (2,5 points)

Un contrôleur gère les deux périphériques suivants :

⇒ 0 : Lecteur de CDROM ;

⇒ 1 : Lecteur de disque dur ;

Pour cela, le contrôleur utilise deux canaux :

⇒ Canal 0 pour le lecteur de disque dur.

⇒ Canal 1 pour le lecteur de disquette.

Chaque canal contient les registres suivants :

⇒ RE_i: Registre d'Etat.

⇒ RC_i: Registre de Commandes,

⇒ RD_i: Registre de Données,

⇒ RS_i: Registre de Secteur,

⇒ RP_i: Registre de Piste,

Le contrôleur exécute les commandes suivantes :

⇒ SEEK (Recherche piste) → code opération: "001"

⇒ Read (Lecture secteur) → code opération: "010"

⇒ WRITE (Ecriture secteur) → code opération: "011"

⇒ STOP (Fin du programme d'entrée/sortie) → code opération: "000"

L'unité centrale accède aux registres des canaux du contrôleur en utilisant les instructions suivantes:

⇒ LDI val : chargement immédiat de l'accumulateur avec la valeur "val".

⇒ LDA RD_i : chargement du contenu du RD du canal i dans l'accumulateur.

⇒ LDA RE_i : chargement du contenu du RE du canal i dans l'accumulateur.

⇒ LDA adr : chargement direct de l'accumulateur avec la valeur contenue dans le mot adr.

⇒ LDA *, adr : chargement de l'accumulateur avec la valeur contenue dans le mot pointé par l'adresse qui se trouve dans le mot mémoire adr.

⇒ STA RD_i : rangement de l'accumulateur dans le RD du canal i.

⇒ STA RC_i : rangement de l'accumulateur dans le RC du canal i.

⇒ STA RS_i : rangement de l'accumulateur dans le RS du canal i.

⇒ STA RP_i : rangement de l'accumulateur dans le RP du canal i.

⇒ STA adr : rangement direct du contenu de l'accumulateur dans le mot adr.

⇒ STA *, adr : rangement du contenu de l'accumulateur dans le mot pointé par l'adresse qui se trouve dans le mot mémoire adr.

⇒ BZ étiquette : branchement conditionnel à l'étiquette étiquette si le contenu de l'accumulateur est nul.

⇒ BNZ étiquette : branchement conditionnel à l'étiquette étiquette si le contenu de l'accumulateur n'est pas nul.

⇒ BI étiquette : branchement inconditionnel à l'étiquette étiquette.

⇒ **ADDI val**: addition en mode immédiat de la valeur "val" au contenu de l'accumulateur; résultat dans l'accumulateur. Cette instruction positionne les indicateurs.
 ⇒ **SUBI val**: soustraction en mode immédiat de la valeur "val" au contenu de l'accumulateur; résultat dans l'accumulateur. Cette instruction positionne les indicateurs.
 ⇒ **ANDI 'valeur binaire'**: et logique en mode immédiat entre le contenu de l'accumulateur et la valeur binaire précisée dans le champ opérande; résultat dans l'accumulateur. Cette instruction positionne les indicateurs.

Toutes les opérations doivent se faire dans l'accumulateur. L'unité centrale ne peut faire aucun test directement sur les registres internes du contrôleur. Le contenu du registre à tester doit d'abord être chargé dans l'accumulateur.

Les bits des registres d'état RE_i des deux canaux sont positionnés par le contrôleur pour indiquer :

- ⇒ $B0_i=1$ { Le Registre de Données du canal i est plein en lecture.
 { Le Registre de Données du canal i est vide en écriture.
- ⇒ $B1_i=1$ La commande exécutée sur le périphérique relié au canal i est terminée.
- ⇒ $B2_i=1$ Le périphérique et le canal i sont prêts.
- ⇒ $B3_i=1$ Indicateur général d'erreur (erreur sur l'entrée/sortie).
- ⇒ $B4_i=1$ Erreur de formatage sur le disque (code erreur: 00).
- ⇒ $B5_i=1$ Erreur d'écriture sur le disque (code erreur: 01).
- ⇒ $B6_i=1$ Erreur de lecture sur le disque (code erreur: 10).
- ⇒ $B7_i=1$ Erreur Numéro de piste inexistant (code erreur: 11).

Question:

Ecrire le programme exécuté par l'unité centrale, qui fait la lecture de 256 octets du secteur 7 de la piste 3 du CDROM et les copie dans une zone mémoire qui commence à l'adresse 1029 en mémoire centrale.

Remarques :

- Vous devez utiliser seulement les instructions qui sont décrites dans le sujet.
- Le programme doit se terminer par une commande STOP.
- On ne tiendra pas compte des anomalies (erreurs durant les différentes opérations). On supposera qu'aucune erreur ne se produira durant les différentes opérations.

Exercice 3 : (3,5 points)

Soit le système d'interruptions hiérarchisé d'un ordinateur. Il est possible de valider ou invalider le système d'It, Masquer un niveau et inhiber une cause. Il permet de recevoir 12 causes d'interruption réparties comme suit :

Niveau 0:	* Cause 0 : It alimentation;
Niveau 1:	* Cause 1 et cause 2 : It contrôleurs d'entrées/sorties;
	* Cause 3 : It inhibée;
	* Cause 4 : It Inhibée;
Niveau 2:	* Cause 5 : It contrôleur d'Entrées/Sorties;
	* Cause 6 : It contrôleur d'Entrées/Sorties;
	* Cause 7 : It matériel;
	* Cause 8 : Inhibée;
Niveau 3:	* Cause 9 : It contrôleur d'Entrées/Sorties;
	* Cause 10 : Inhibée;
	* Cause 11 : Inhibée;

Soit la séquence d'interruptions suivante :

- Allumage de la machine ;
- Lancement d'un programme de niveau 4 ;
- Signal de niveau 3 cause 9 pendant l'exécution du programme ;
- Arrivée d'un Signal de niveau 2, cause 6 pendant la phase précédente ;
- Arrivée d'un Signal de niveau 0, cause 0 pendant la phase précédente ;

Questions :

1. Faire un schéma détaillé du système d'It **en indiquant le contenu des registres importants au démarrage de la machine.**

Remarque importante :

Un Niveau 4 existe, mais il est réservé aux interruptions internes (ce niveau ne doit pas apparaître sur le schéma mais un bit lui sera réservé dans le registre masque).

2. Représenter sur un schéma la séquence en donnant à chaque étape une étiquette (Instant A, Instant B...). Vous devrez identifier clairement chacune des étapes importantes de la séquence.
3. Indiquer le contenu de la pile et du registre masque à chaque instant en incluant le niveau 4 dans le registre masque.