

UNIVERSITE BLAISE PASCAL
CLERMONT-FERRAND II

Service des concours
34 avenue Carnot
63000 Clermont-Ferrand

Concours **EXTERNE** - BAP E
Corps : **Assistant Ingénieur**
Spécialité : **Développeur, intégrateur
d'applications**

Session **2009**

Epreuve écrite d'**Admissibilité**

Durée : **3 heures** - Coefficient : **4**

Le sujet comporte 6 pages numérotées de 1 sur 6 à 6 sur 6.

Assurez-vous que cet exemplaire est complet. S'il est incomplet, demandez un autre exemplaire au surveillant de salle.

Il vous est rappelé que votre identité ne doit figurer que sur la première page de la copie d'examen. Toute mention d'identité portée sur toute autre partie de la copie que vous remettrez en fin d'épreuve mènera à l'annulation de votre épreuve.

L'usage de la calculatrice est interdit.

L'usage du téléphone portable est interdit.

Tout document et autre matériel électronique sont interdits.

Ne pas écrire au crayon de papier sur la copie.

PARTIE I - Algorithmique / Programmation

1. Pointeurs en C

Rappel sur les pointeurs en C :

L'opérateur + permet de réaliser la somme de deux valeurs arithmétiques, mais il permet également de réaliser la somme d'un pointeur et d'un entier.

Une telle opération n'a de sens cependant, que si le pointeur repère un élément d'un tableau.

Soient p une valeur pointeur vers des objets de type T et un tableau dont les éléments sont du même type T . Si p repère le $i^{\text{ème}}$ élément du tableau, $p + j$ est la valeur de type pointeur vers T , qui repère le $(i + j)^{\text{ème}}$ éléments du tableau (en supposant qu'il existe).

Il en va de même avec l'opérateur soustraction, et si p repère le $i^{\text{ème}}$ élément d'un tableau, $p - j$ repère le $(i - j)^{\text{ème}}$ élément du tableau (toujours en supposant qu'il existe).

Soit les déclarations suivantes :

```
#define N 10
int t[N];
int *p, *q, *r, *s;
```

```
p = &t[0];
q = p + (N-1);
```

```
r = &t[N-1];
s = r - (N-1);
```

Indiquer ce que représentent p , q , r et s .

Soit un tableau d'entiers t . Ecrire une procédure main qui parcourt le tableau t et qui imprime les indices des éléments nuls du tableau, **sans utiliser aucune variable de type entier**.

```
#define N 10
int t[N]={1,2,0,11,0,12,13,14,0,4};
```

2. Programmation Objet

- Principes généraux

Définir les concepts de classe et d'objet, décrire le lien entre ces 2 notions.

Comment les objets communiquent-ils entre eux ?

Qu'est-ce que l'héritage ?

Qu'est-ce que le polymorphisme ?

- Exercice

On désire ajouter à une télécommande une fonction qui lui permettra d'allumer tous les appareils électriques d'une maison. On suppose que la télécommande dispose d'une liste des appareils à allumer. Sachant que chaque appareil dispose d'une commande pour s'allumer et en tenant compte du fait que la manière dont il s'allume dépend de l'appareil, utiliser les concepts d'héritage et de polymorphisme pour concevoir les classes nécessaires à la mise en œuvre.

On se limitera dans un premier temps à 2 catégories d'appareil : une télévision et un four.

Ecrire le code de la classe Télécommande en Java.

- Design Pattern

Design pattern Singleton

President
<u>-instance: President</u> -nom
-President() <u>+getInstance(): President</u> +getNom() +setNom(nom)

Soit la classe Président modélisée ci-dessus. On a appliqué à cette classe le design pattern singleton. Indiquer quelle est l'utilité de ce design pattern et comment il fonctionne.

Design pattern Composite

Le design pattern composite permet d'assembler des objets dans des structures arborescentes tout en permettant au client de traiter les objets individuels et les compositions de la même manière. Modéliser une arborescence de Répertoires et de Fichiers en utilisant (ou retrouvant) ce design pattern. Décrire brièvement l'algorithme de parcours d'un répertoire.

PARTIE II – Base de données SQL

Soit les 4 tables d'une base de données décrites ci-dessous :

Fournisseur(NF, nomF, statut, ville)

Produit(NP, nomP, poids, couleur)

Usine(NU, nomU, ville)

Livraison(NP, NU, NE, qt)

1. Donner les instructions de création des tables de cette base.
2. Elaborer les ordres SQL correspondant aux requêtes suivantes :
 - Liste des couleurs triées par ordre alphabétique décroissant
 - Ensemble des numéros de producteur de produits rouges
 - Combien de produits différents a livré le fournisseur "SARL"
 - Combien de produits différents ont été livrés par chacun des fournisseurs tels que la quantité totale de produit livré par ce fournisseur soit supérieur à 1000

PARTIE III – Modélisation

Il s'agit de modéliser le problème classique appelé producteurs/consommateurs. De manière indépendante et aléatoire plusieurs producteurs produisent des produits et plusieurs consommateurs viennent les chercher pour les consommer.

Proposer une **modélisation OBJET** de ce problème pour que cela fonctionne avec n'importe quel type de produit.

PARTIE IV - Questions diverses

1. Qu'est ce qu'une architecture 3 tiers ?
2. Commenter en une ligne chacun des sigles suivants : XML, UML, SAX, DOM, DTD, SOAP.
3. Que signifie CSS, quel est son apport dans la conception des pages html ?

4. Soit le code :

```
<html>  
  <form action="http://220.7.51.91/Appli/HTM/AfficheNom" method="get">  
    <head>  
      <link rel="stylesheet" type="text/css" href="../CSS/APPLI.CSS">  
      <title>Ecran d'identification</title>  
    </head>  
    <h1 align="center">Identification</h1>  
    <body>  
      <p/>  
      <p/>  
      <h3>  
        Utilisateur &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input name="USR" type="text"  
value="">  
      </h3>  
      <p/>  
      <h3>  
        Mot de pass  <input name="MDP" type="password" value="">  
      </h3>  
      <p/>  
      <h1 align="center">  
        <input type="submit" align="middle" value="Valider">  
      </h1>  
    </body>  
  </form>  
</html>
```

Que se passe-t-il lorsqu'on appuie sur le bouton "Valider" ? Expliquer pourquoi ? Que veut dire method="get" ?