

Examen #2

INF3135 groupe 20

Date: Jeudi, 4 août 2005.

Durée: 3 heures.

Nom: _____
Code permanent: _____

/	20
/	20
/	30
/	30
/	100

Question #1: (20 pts) Programmation défensive et assertion.

a) (10 pts) Expliquez en quoi les principes de justesse et de robustesse sont contradictoire.

b) (10 pts) Voici une structure de données :

```
typedef struct {  
    double x, y;  
} Point;
```

soit l'entête de routine suivante :

```
// distance2  
// entrees :  
//   Point *p1  
//   Point *p2  
// sorties :  
//   La distance au carre entre p1 et p2.  
//   distance = ( p1.x - p2.x ) * ( p1.x - p2.x ) + ( p1.y - p2.y ) * ( p1.y - p2.y )  
double distance2( Point *p1, Point *p2 );
```

Écrivez le code (en langage C) pour cette routine avec les assertions nécessaires pour respecter le contrat.

Question #2: (20 pts) Stratégies de tests et tests unitaires.

a) (10 pts) Les tests *béta* sont effectués chez le client. Cette pratique risque de montrer une version peu fiable au client. Pourquoi est-il quand même important d'effectuer des tests *béta*.

b) (10 pts) L'approche de développement dirigé par les tests demande d'écrire les tests avant le code. Quelles sont les avantages d'une telle pratique ?

Question #3: (30 pts) Conception de logiciels.

a) (10 pts) Le problème de la conception de logiciel n'accepte pas de solution parfaite, seulement des solutions bonnes ou mauvaises. Expliquez ce qu'est une bonne conception de logiciel.

b) (10 pts) Quels sont les désavantages d'un couplage fort entre deux modules ?

c) (10 pts) La construction d'un module demande la création d'une interface (fichier .h) et d'une mise en oeuvre (fichier .c). Il est possible de diviser le processus de conception en deux étapes: la conception architecturale et la conception détaillée. Est-il correct d'affirmer que la création d'interface se situe dans l'étape de conception architecturale et que la construction de mise en oeuvre se situe dans l'étape de conception détaillée ? (Expliquez votre réponse.)

Question #4: (30 pts) Étude de cas.

Voici une section du fichier Sequences.h :

```
typedef void * Element;

struct liste
{
    ...
};

typedef struct liste Liste;

typedef struct
{
    ...
} Sequence;

Sequence *creerSequence();
void detruireSequence (Sequence *s);
void ajouter (Sequence *s, Element e);
void retirerKieme (Sequence *s, int k);
int nbElements (Sequence *s);
Element kiemeElement (Sequence *s, int k);
void initGénérateur_Seq (Sequence *s);
int prochainElement_Seq (Sequence *s, Element *e);

/* exemple d'utilisation de generateur pour
   parcourir la sequence: La sequence sera
   parcourue a partir de l'element de tete,
   vers l'element de queue.

   Sequence *seq;
   Element e;
   ...
   initGénérateur_Seq(seq);
   while(prochainElement_Seq(seq, &e))
   {
       ...
   }
*/
```

a) (4 pts) Le module Sequences est de quel type :

A - Bibliothèque de routines.

B - Machine abstraite.

C - Collection de valeur.

D - Classe d'objet.

b) (9 pts) Quel est le niveau de couplage pour les routines suivantes:

i) (3 pts) ajouter :

ii) (3 pts) retirerKieme :

iii) (3 pts) prochainElement_Seq :

c) (9 pts) Quel est le niveau de cohésion pour les routines suivantes:

i) (3 pts) ajouter :

ii) (3 pts) retirerKieme :

iii) (3 pts) prochainElement_Seq :

d) (8 pts) Si le client appelle la routine retirerKieme avec une valeur entière plus grande que la taille de la séquence, devrions-nous utiliser une assertion ou une erreur pour détecter le problème ? Expliquez votre réponse.