**Contrôle Continu AP4A – Mardi 14 Mai 2024 – durée 1h**

**I.** (4pts) Indiquer quelles sont les erreurs de compilation lorsqu’il en existe dans les exemples C++ suivants. Dans tous les cas de figure, expliciter la réponse.

**1)**

class B {

protected:

 int i;

};

class D : public B {};

void f() {

 B\* p = new B;

 D\* q = new D;

 int fi1 = p->i;

 int fi2 = q->i;

}

**2)**

class A {

protected:

 int a;

};

class B : public A {void f();};

class C : private B {void f();};

void B::f() {a = 1;}

void C::f() {a = 1;}

**3)**

class A {

protected:

 int a;

};

class B : private A {void f( );};

class C : public B {void f( );};

void B::f() {a = 1;}

void C::f() {a = 1;}

**4)**

class A {

protected:

 int a;

};

class B : protected A {void f();};

class C : private B {void f();};

void B::f() {a = 1;}

void C::f() {a = 1;}

**II.** (6 pts) Schéma composite. On veut modéliser en C++ la notion de composant graphique composite. Une classe de base ObjetGraphique représente un objet graphique abstrait qui peut être affiché dans une console par la méthode abstraite afficher(). Un objet graphique possède une position (*x*,*y*). Un cercle est un objet graphique particulier possédant un rayon *r*. Un rectangle est un objet graphique et possède une hauteur *h* et largeur *l*. Enfin, un objet graphique composite est un ensemble d’objets graphiques qui peuvent être des cercles, des rectangles ou des composites.

1) Donner une définition complète (interface et corps) des classes ObjetGraphique, Cercle, Rectangle, et Composite, préciser bien les constructeurs et un exemple de fonction afficher(), de façon que le programme ci-dessous fonctionne et affiche les caractéristiques d’objets graphiques.

main (){

 ObjetGraphique \*c = new Cercle(0, 0, 10) ;

 ObjetGraphique \*r = new Rectangle(0, 0, 20, 30) ;

 ObjetCompose \*oc = new ObjetCompose() ;

 (\*oc).ajoute(c) ;

 (\*oc).ajoute(r) ;

 ObjetCompose \*oc2 = new ObjetCompose() ;

 (\*oc2).ajoute(c) ;

 (\*oc2).ajoute(r) ;

 (\*oc).ajoute(oc2) ;

 oc->afficher() ;

}

2) Faut-il définir la forme de Coplien pour la classe ObjetCompose ?

**III.** (5 pts) Soit une classe Animal dont on n’impose pour l’exercice que le code suivant :

 class Animal {String nom; public: ... } ;

 main() {

 Animal\* ptr[3] ;

 ptr[0] = new Chat("miaou") ;

 ptr[1] = new Oiseau("pioupiou") ;

 ptr[2] = new Chat("minou") ;

 for (int i=0;i<3;i++) {

 ptr[i]->affiche() ;

 }

 }

1) Définir les classes (interface et corps) Animal, Chat et Oiseau pour que le résultat du main soit le suivant :

 Chat miaou

 Oiseau pioupiou

 Chat minou

2) La classe Animal doit-elle vérifier la forme de Coplien ? Pourquoi ?

3) Réécrire le main dans le cas où on utilise un tableau de valeurs et non pas de pointeurs ?

 main() { Animal ptr[3]; ... }

**IV.** (5 pts) Classe FIFO. On dispose d'une classe Liste d’éléments de type T telle que vue en TD. On veut définir une classe file FIFO (First-In-First-Out) à partir de la classe Liste. Les opérations autorisées pour cette file sont les suivantes. L'opération “ajoutQueue” ajoute un élément à la fin de la file, l'opération “supprimeTete” supprime la tête de la file et renvoie cette dernière, l’opération « tete » renvoie la tête de la file et l’opération estVide() renvoi « true » si la file est vide, « false » sinon.

Donner une implémentation de la classe File par héritage privé.