

**Informations générales**

Examen de module	Examen écrit	Date : 23.09.2021
Nom du module : Programmation orientée objet	Numéro de module :	40050200
Durée de l'examen : 60 min + 15 min	Examineur : Welp	

Nom : _____ Prénom _____

Numéro de matricule : _____

Évaluation

Tâche	1	2	3	4						Total
Points à atteindre	11	13	15	15						54
Points obtenus										
Signature de l'examineur				Points obtenus en					Note	

Tâche 1: Du C au C++

Soit le type de données suivant :

```
struct ComplexT{  
    double r,i ;  
} ;
```

1. Ecrivez une fonction `phase` pour calculer la phase d'un nombre complexe. Le nombre complexe doit être passé via un paramètre `z` de type `ComplexT`. La fonction doit retourner la phase en degrés ou en radians, au choix. Ceci doit être contrôlé par un deuxième paramètre `unité` de type `int` (0 : radian, 1 : degré). L'indication de l'unité doit être facultative lors de l'appel de la fonction. Si elle est omise, le calcul doit être effectué en radians.

Résolvez ce problème en

1. `phase` surchargée
2. définir la `phase` comme une fonction avec des paramètres par défaut.

Donnez les définitions des fonctions pour les deux solutions. (7 points)

Remarque : pour calculer la phase, on peut utiliser la fonction `atan2` de la bibliothèque standard C-.

2. Définissez une fonction `inverser` qui inverse le signe de la partie réelle d'une variable `ComplexT` passée en **référence**. Définissez également une fonction `main` dans laquelle la fonction `Inverser` est testée. (4points)

Tâche 2 : Classes et objets

Définissez une classe `LieuVec` pour stocker et calculer les vecteurs de lieu avec les propriétés suivantes (*NB : laissez suffisamment de place dans la classe pour pouvoir faire des ajouts au fur et à mesure*) :

- L'accès direct aux attributs d'un vecteur ne doit pas être possible. L'accès doit se faire via les méthodes `set` et `get` correspondantes. Implémentez les méthodes comme des fonctions en ligne.
- La classe doit disposer de constructeurs appropriés pour l'initialisation des objets `Vec locaux`, qui permettent une instanciation de l'objet conformément à la remarque 1. Lors de l'instanciation d'un objet `Vec local` sans paramètre, les composants du vecteur doivent être initialisés avec 0. Implémentez les méthodes comme des fonctions en ligne.
- Une méthode doit permettre de demander la longueur d'un objet `LieuVec`. Implémentez cette méthode en dehors de la définition de la classe. Dans le fragment de programme de la remarque 1, insérez une instruction permettant d'obtenir la longueur du vecteur `v1`.
- La classe doit disposer d'un attribut de classe dans lequel le nombre d'objets `Vec locaux` est enregistré (compteur d'objets). Le nombre d'objets `Vec locaux` doit pouvoir être interrogé via une méthode de classe.
- De plus, l'opérateur `^` doit être surchargé pour la classe `LieuVec`. L'opérateur `^` doit déterminer le produit scalaire de deux vecteurs de lieu. **Déclarez** l'opérateur. (*voir note*)

(13 points)

Remarques

- La classe `Vecteur` doit par exemple pouvoir être utilisée de la manière suivante

```
int main()
{
    LieuVec v1(1,2,3),v2 ;
    v2.setX(4) ;
    v2.setY(5) ;
    v2.setZ(6) ;

    cout << "Le produit scalaire de v1 et v2 est : " << v1^v2 << endl ;
    cout << "Il y a en tout " << Vecteur::getNombre() << " vecteurs" ;
}
```

édition :

```
Le produit scalaire de v1 et v2 est
: 32 Il y a en tout 2 vecteurs.
```

- La longueur d'un vecteur est calculée à partir de $\sqrt{x^2 + y^2 + z^2}$

Tâche 3: Relations entre objets et modèles

Dans le logiciel de bibliothèque d'une université, il existe les classes `étudiant` et `livre`. Entre les objets de ces deux classes, il existe l'association suivante ("l'étudiant réserve le livre") :



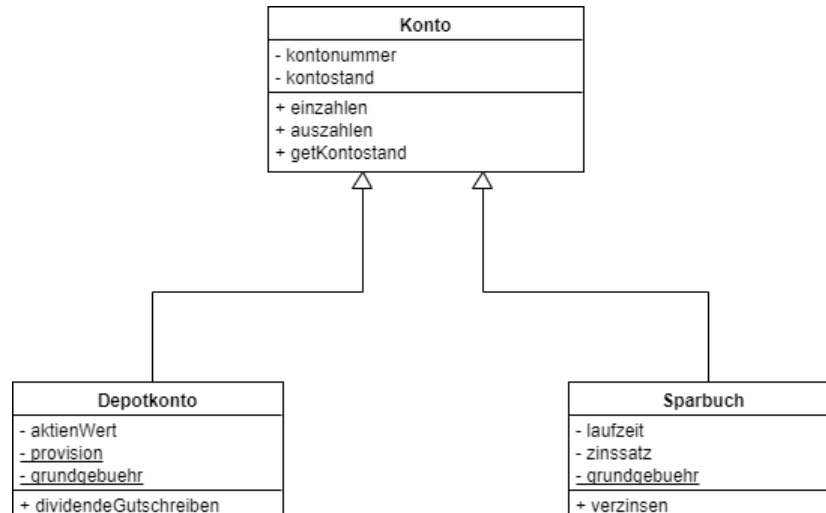
```
class Student{
    string name;
public:
    Student(string n):name(n){};
    string getName(){return name;};
};
```

```
class Buch{
    string titel;
public:
    Buch(string t):titel(t){};
    string getTitel(){return titel;};
};
```

1. Complétez la définition de la classe `Étudiant` de manière à ce que l'association avec la classe `Livre` soit mise en œuvre. Il doit exister des méthodes permettant de réserver un livre, d'annuler une réservation de livre et de consulter les réservations. Décrivez brièvement la tâche, les paramètres et la valeur de retour des méthodes ! *Les méthodes n'ont pas besoin d'être implémentées.* (8 points)
2. Ecrivez un programme principal dans lequel une variable conteneur est définie pour stocker des objets étudiants. **Pour ce faire, utilisez une classe conteneur de la Standard Template Library (STL).** (1points)
Ajoutez quelques (au moins 3) objets étudiants à la variable. (2points)
Ajoutez à votre programme des instructions permettant d'afficher le nom de tous les étudiants stockés. (4 points)

Tâche 4: Héritage et polymorphisme

1. Soit le diagramme de classes suivant :



La classe `Compte` est définie comme suit :

```
classe Compte
{
    unsigned int numéro de
    compte ; float solde de
    compte ;
    public :
        Compte(unsigned int knr, float ks=0.0.) ;
        float getSoldeCompte() ;
        void déposer(float b) ;
        void payer(float b) ;
} ;
```

- a) Donnez une définition de classe pour la classe `Compte` de dépôt. (4 points)
- b) De quels attributs disposent **les objets** de la classe `Compte` de dépôt ? (3 points)
- c) De quelles méthodes disposent **les objets** de la classe `Compte` de dépôt ? (2 points)
2. Quelles sont les affirmations concernant le modèle de conception d'Observer que vous approuvez ? (2 points)

- ☐ Lorsque l'interface d'un sujet change, cela a également des répercussions sur l'observateur.
- ☐ Avant qu'un observateur ne soit informé des changements d'état d'un sujet, il doit s'enregistrer auprès du sujet.
- ☐ Les sujets actualisent leur état dès que des observateurs s'enregistrent auprès des sujets.
- ☐ Un observateur appelle sporadiquement la méthode `notify()` du sujet afin d'être informé des changements d'état du sujet.

- ☐ L'observateur doit implémenter une interface définie dans une classe de base abstraite à partir de laquelle l'observateur est dérivé.