

**Informations générales**

Examen de module	Examen écrit	Date : 03.02.2022
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	8	13	9	19						49
Points obtenus										
Signature de l'examineur				Points obtenus en					Note	

Tâche 1: Du C au C++

1. Ecrivez une fonction `valeur` pour calculer la valeur d'un terrain à bâtir. La valeur est calculée à partir de la surface du terrain x 235,-EUR et de la surface habitable du bien immobilier x 2530,-EUR. La fonction doit recevoir comme paramètres la surface du terrain et, en option (si le terrain est déjà construit), la surface habitable.

Résolvez ce problème en

1. `valeur` surchargée
2. en tant que fonction avec des paramètres par défaut.

Donnez les définitions des **fonctions** pour les deux solutions. (5 points)

2. Qu'est-ce que le fragment de programme suivant affiche à l'écran ? (3 points)

```
long n[3] = {2000,3000,4000};  
long& r = n[1];  
long* p = n;  
r = r/10;  
p=p+2;  
*p = r + n[0]/2;  
cout << n[2] << " " << r << " " << *p << endl;
```

Tâche 2: Classes et objets

Définissez une classe `Fraction` pour la représentation de fractions. Le numérateur et le dénominateur doivent être stockés dans la classe en tant qu'attributs entiers. La classe doit posséder les propriétés suivantes :

- L'accès direct au numérateur et au dénominateur d'une fraction ne doit pas être possible. L'accès doit se faire via les méthodes `setter` et `getter` correspondantes. **Implémentez** les méthodes comme des fonctions en ligne.
- La classe doit disposer de constracteurs appropriés pour l'initialisation d'objets fractionnaires. Un objet fractionnaire doit pouvoir être initialisé par l'indication d'un numérateur et d'un dénominateur ainsi que par un nombre entier. L'instanciation d'un objet sans paramètre doit également être possible. Dans ce cas, l'objet doit être initialisé avec la valeur 0 (voir aussi la remarque). **Implémentez** les méthodes comme des fonctions en ligne.
- La classe doit mettre à disposition une méthode `getDouble()` qui renvoie la fraction sous forme de valeur double. **Implémenter** la méthode en dehors de la classe.
- De plus, l'opérateur `~` à un chiffre pour la classe `fraction` doit être surchargé de manière à ce que la fraction puisse être réduite par le plus grand diviseur commun du numérateur et du dénominateur. La fonction `int ggt(int a, int b) ;` est disponible. **Implémentez** la méthode en dehors de la classe. (13 points)

Remarque

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

```
int main()
{
    Fraction
    b1(12,9) ;
    Fraction b2(5)
    ;
    Fracture b3 ;

    cout << "b1 = " << b1.getZaerhler() << "/" << b1.getNenner()
        << " = " << b1.getDouble() << endl ;
    cout << "b2 = " << b2.getZaehler() << "/" << b2.getNenner()
        << " = " << b2.getDouble() << endl ;
    cout << "b3 = " << b3.getZaerhler() << "/" << b3.getNenner()
        << " = " << b3.getDouble() << endl ;

    b3 = ~b1 ;
    cout << "b1 = " << b1.getZaehler() << "/" << b1.getNenner()
        << " = " << b1.getDouble() << endl ;
    cout << "b3 = " << b3.getZaehler() << "/" << b3.getNenner() << " = "
        << b3.getDouble() << endl ;
}
```

édition :

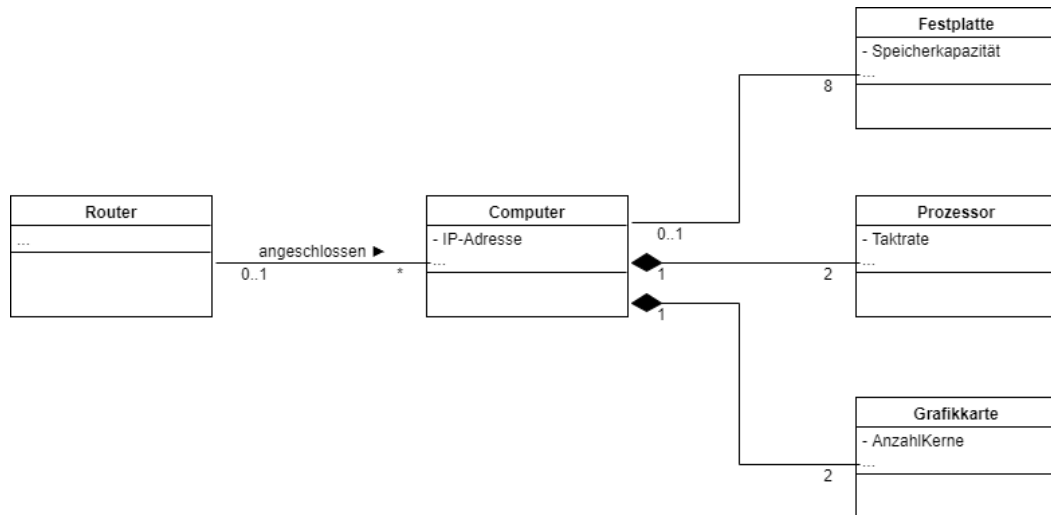
```
b1 = 12/9 = 1.33333
b2 = 5/1 = 5
b3 = 0/1 = 0
b1 = 4/3 = 1.33333
```



$b_3 = 4/3 = 1.33333$

Tâche 3 : Relations entre objets et modèles

Dans un logiciel de gestion de réseau, il y a les classes `routeur`, `ordinateur`, `disque dur`, `processeur` et `carte graphique`, qui sont en relation les unes avec les autres selon le diagramme de classes suivant.



1. Quels attributs sont nécessaires dans la classe `Ordinateur` pour implémenter les relations avec les classes `Routeur`, `Disque dur`, `Processeur` et `Carte graphique` ? Répondez à la question en donnant la définition de classe pour la classe `Ordinateur` (attributs uniquement, les méthodes ne sont pas nécessaires). (4 points)
2. **Définissez** une **fonction de modèle** `maximum` qui renvoie la plus grande valeur d'un tableau. Les éléments du tableau doivent être de n'importe quel type. (5 pts)

application possible :

```

long a[4] = {2,5,3,4} ;
chaîne s[3] = {"Bochum", "Dortmund", "Essen"} ;
cout << maximum(a,4) << endl ;
cout << maximum(s,3) << endl ;
  
```

édition :

5

Repas

Tâche 4 : Héritage et polymorphisme

Soit les deux classes suivantes pour la gestion des terrains résidentiels et des terrains commerciaux. Un impôt foncier est prélevé sur les deux types de terrains, calculé différemment pour les terrains résidentiels et commerciaux. PAR EXEMPLE

$$\text{grundsteuer}(\text{wohngrundstück}) = (\text{flaeche} * 10 + \text{wohnflaeche} * 300) * \frac{\text{zinssatz}}{100}$$

$$\text{grundsteuer}(\text{gewerbegrundstueck}) = \text{flaeche} * \text{nutzungsgrad} * \frac{\text{zinssatz}}{100}$$

Le taux d'intérêt est différent pour les terrains résidentiels et commerciaux.

Wohngrundstueck	
- flaeche	/* in m^2 */
- wohnflaeche	/* in m^2 */
- zinssatz	/* in % */
+ berechneGrundsteuer	
+ ausgabe	
+ <u>setZinssatz</u>	

Gewerbegrundstueck	
- flaeche	/* in m^2 */
- nutzungsgrad	/* [1...100] */
- zinssatz	/* in % */
+ berechneGrundsteuer	
+ ausgabe	
+ <u>setZinssatz</u>	

1. Effectuez une refonte dans le but d'externaliser les points communs dans une classe de base en utilisant l'héritage. Esquissez le nouveau diagramme de classe. *Réfléchissez bien à ce qui est vraiment commun.* (4 points)
2. Donnez une définition de classe pour au moins une classe dérivée. (5 points)
3. Comment faire de la méthode `calculerTaxes` de base une méthode purement virtuelle ? (2 points)
4. De quels attributs disposent les objets de la classe dérivée ? (3 points)
5. En quoi l'implémentation suivante de la méthode `setZinssatz` serait-elle incorrecte ? (2 points)

```
void Terrain d'habitation::setTaux d'intérêt(double zs)
{
    if(surface>1000) taux
        d'intérêt=zs*2 ;
    else
        taux d'intérêt=zs ;
}
```

6. **Qt** : Supposons que vous souhaitiez qu'un son soit émis lorsque vous cliquez avec le bouton de la souris dans la fenêtre de votre programme. Quelles dispositions devez-vous prendre dans votre classe de fenêtre (widget) pour que cela soit possible ? (3 points) *Plusieurs réponses possibles.*

- ☐ Votre classe doit être dérivée de `QSound`.
- ☐ Votre classe a besoin d'un attribut du type d'une classe `QWidget` (par ex. `QWidget`). Vous l'utilisez pour appeler la méthode `mousePressEvent`.

- ☐ Votre classe doit être dérivée d'une classe de Qt-Widget (par exemple `QWidget`).

- ☐ Vous devez implémenter une méthode `onMouseClicked`.
- ☐ Vous devez redéfinir la méthode virtuelle `mousePressEvent`.
- ☐ Votre classe doit implémenter un slot qui réagit à un événement de souris.