

Partiel

Lundi 14 Novembre 2011

Aucun document n'est autorisé. Les téléphones portables, comme tout autre moyen de communication vers l'extérieur, doivent être éteints. Le temps à disposition est d'une heure. Motivez bien vos réponses.

1 Exercice [2 points]

Soit la fonction de prototype suivant :

```
void f(int *c);
```

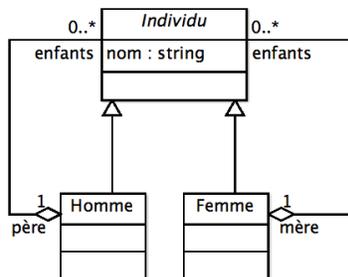
et les déclarations de variables suivantes :

1. `int *pc1;`
2. `const int *pc2;`
3. `char *int pc3;`
4. `const int *const pc4;`

Indiquez en motivant rapidement (mais clairement) lesquelles de ces variables peuvent ou ne peuvent pas être employées lors d'un appel (sans conversion) à la fonction `f`.

2 Exercice [5 points]

Soit le diagramme UML suivant (on rappelle que les noms en italique désignent des abstractions):



Écrire en C++ les classes qui correspondent au diagramme précédent avec les accesseurs (des attributs et relations et les constructeurs).

3 Exercice [3 points]

On suppose avoir réalisé les constructions de l'exercice précédent. Dans le code suivant, expliquez ce qui ne peut fonctionner, pourquoi et comment le corriger:

```
void laDescendance(Individu quidam) {
    cout << quidam.getNom() << "(" << endl;
    for (int i=0; i<quidam.nombreDEnfants(); i++) {
        laDescendance(quidam.enfant(i));
    }
    cout << ")" << endl;
}
int main() {
    Homme h("un homme");
    Femme f("une femme");
    Femme f2("une autre femme");
    Homme h1("jacques",h,f);
    Femme f1("annie",h,f);
    Homme h2("averell",h1,f2);
    Individu &quelquun = h;
    laDescendance(quelquun);
}
```

4 Exercice [5 points]

On désire implémenter un type chaîne de caractères permettant d'éviter les caches à l'utilisateur la gestion de la mémoire héritée du C. Écrire une classe `Chaîne` qui permettrait d'écrire le code suivant :

```
Chaîne s = "Bonjour";
Chaîne s2;
s2 = s;
s2[3] = 'J';
cout << s << endl;
cout << s2 << endl;
```

lequel produirait à l'écran :

```
Bonjour
BonJour
```

5 Exercice [5 points]

Soit le bout de programme C++ suivant correspondant à la simulation d'un jeu de morpion (tour à tour chaque joueur choisit une case et s'il possède trois cases alignées il gagne) :

```

void doIt(Partie p) {
    p.demarre();
    for (int i=0; i<2; i++) {
        if (p.joueur(i)) {
            cout << p.joueur(i).getNom() << " a gagné" << endl;
        } else {
            cout << p.joueur(i).getNom() << " n'a pas gagné" << endl;
        }
    }
}

int main() {
    Joueur j1("luke la main froide");
    Joueur j2("dédé la sardine");
    Plateau p(3,3); // le plateau de jeu est 3x3
    Morpion partie(j1,j2,p);
    doIt(partie);
}

```

On suppose disposer de fonctions auxiliaires :

- `int estPlein(...)`; qui permet de déterminer s'il reste ou non des cases à choisir;
- `int estCorrecte(...)`; qui permet de déterminer si une case choisie est libre ou non;
- `int estUnAlignement(...)`; qui permet de déterminer si la case choisie correspond à un alignement.

Écrire le code de la méthode `demarre()`; de la classe `Morpion` ainsi que les méthodes éventuellement nécessaires des autres classes concernées.