

## Contrôle de connaissances INF224 – Février 2016

**Durée: 1h.**

Documents et ordinateur **non autorisés** (dictionnaire papier autorisé pour les non-francophones).

Les réponses peuvent être en anglais.

SVP écrivez **lisiblement** et **laissez une marge**.

**1. (3 points)** Quels sont les 4 grandes familles de langages de programmation ? Expliquer en très peu de lignes leurs caractéristiques.

**2. (2 points)** Qu'est-ce qu'une méthode d'instance, une méthode de classe, une fonction non-membre ? Quelles sont leurs propriétés respectives et leurs différences ?

**3. (2 points)** Qu'est-ce qu'un smart pointer ? Quelles sont leurs avantages et leurs limitations ? Comment sont-ils implémentés (par exemple `shared_ptr<>`, juste donner le principe).

**4. (2 points)** On souhaite écrire une méthode qui permette de récupérer l'heure et les minutes sous la forme de 2 entiers passés en arguments. On suppose que `foo()` et `calcul()` sont des méthodes de la même classe et que les `...` sont des appels système adéquats. Est-ce que ce qui suit est correct en **C++** et en **Java** ? Corrigez si nécessaire et expliquez.

```
void getTime(int heure, int minutes) {
    heure = ...;
    minutes = ...;
}
void calcul() {
    int heure, minutes;
    getTime(heure, minute);
}
```

### Exercice Java

**5. (4 points)** On veut écrire en **Java Swing** l'interface graphique d'un distributeur de boissons. Ecrire de **deux** manières différentes (i.e. en utilisant des possibilités différentes du langage Java) une classe **Panneau** comportant un bouton **Thé**, un bouton **Café** et un composant qui affichera un message textuel. Lorsque l'utilisateur cliquera sur l'un des boutons, l'afficheur affichera « *En préparation...* » ou « *Choix indisponible* » suivant la disponibilité du produit. Ne mettre que l'essentiel et commentez vos choix.

### Exercice C++ (7 points)

**6. (2 points)** On souhaite créer une librairie de formes géométriques en **C++** que l'on puisse afficher à l'écran via leur méthode **draw()**. Pour ce faire, on déclare une classe abstraite **Shape** dont dériveront diverses sous-classes. On suppose que **Shape** hérite d'une classe **Object** déclarée dans une librairie standard que l'on ne peut pas modifier et qu'il y a les `#include` adéquats. La déclaration ci-après est-elle satisfaisante ? Expliquer.

```
class Shape : public Object {
public:
    void draw();
    float getY();
    float getWidth();
    void setX(float x);
    void setWidth(float x);
    ...
};
```

**7. (3 points)** On veut maintenant définir une classe **Container** qui possède une liste d'**Objects**, certains étant des **Shapes**, d'autres pas. Cette classe aura une méthode `add()` pour y ajouter un **Object** et une méthode `draw()` pour afficher tous les **Objects** qui sont des **Shapes**.

7a) La déclaration ci-après est-elle correcte ?

7b) Implémentez la méthode `add()` et la méthode `remove()`.

Vous pouvez faire toutes les modifications que vous jugez utiles (pensez à lire la question suivante) tant que **Container** gère une liste d'**Objects**. On suppose qu'il y a les `#include` adéquats.

```
class Container {
public:
    string name;
    Object** objects;

    Thing(string _name) : name(_name) {}
    void add(Object*);
    void remove(Object*);
    void draw();
};
```

**8. (2 points)** Cette implémentation de la méthode `draw()` est-elle correcte ? Modifiez si nécessaire.

```
void draw() {
    for (int i = 0; objects[i] != nullptr; ++i)
        objects[i]->draw();
}
```