

Programmation Orientée Objet

2024-2025
Devoir Surveillé

- 4 pages - indiquer le numéro du sujet (1A) sur votre rendu - notes autorisées - calculatrice autorisée - ne pas répondre sur le sujet mais sur une copie à rendre avec nom prénom ou/et num étudiant - par défaut aucune justification n'est demandée - tout appareil électronique communiquant interdit -

Exercice 1. Automaton

On appellera Automaton un automate (déterministe) qui possède 3 états : $\{0, 1, 2\}$ sur un alphabet à 2 lettres $\{a, b\}$. Si l'automaton est dans l'état 0 et qu'on lui applique la transition a il passe dans l'état 2 : on peut résumer la règle ainsi $0 \xrightarrow{a} 2$. Voici l'ensemble des règles :

- $0 \xrightarrow{a} 2$
- $0 \xrightarrow{b} 2$
- $1 \xrightarrow{a} 0$
- $1 \xrightarrow{b} 1$
- $2 \xrightarrow{a} 2$
- $2 \xrightarrow{b} 0$

Écrire la classe `Automaton` qui doit respecter l'exemple d'utilisation donné ci-dessous.

```
1 Automaton x{0}; // construit l'automate x dans l'état 0
2 Automaton y{1}; // construit l'automate y dans l'état 1
3
4 // Automaton z{3};
5 // produit ou signal (de la méthode de votre choix)
6 // une erreur lors de l'exécution car il n'y a pas
7 // d'état 3
8
9 y.trA(); // applique la transition "a" à l'automate "y"
10
11 y.get(); // renvoie l'état actuel de y
12
13 x.trB(); // applique la transition "b" à l'automate "x"
14
15 y.back(); // revient à l'avant dernier état (ne fonctionne qu'une fois)
16 y.back();
17 // sans changement d'état (car on l'a déjà fait à l'instruction
18 // précédente, on ne mémorise que l'avant dernier état)
19 y[1]; // force l'automate à passer dans l'état 1.
20 y.back(); // revient à l'état précédent l'instruction "y[1]"
```

Exercice 2. *Surcharge*

Le fichier C++ suivant compile sans produire d'erreur.

1. Indiquez ce qu'il affiche lors de son exécution.
2. Si on décommente la ligne commentée **L1** du code ci-dessous alors cela produit-il une erreur? Si oui pourquoi? Si non quel sera l'affichage de cette ligne?
3. Si on décommente la ligne commentée **L2** du code ci-dessous alors cela produit-il une erreur? Si oui pourquoi? Si non quel sera l'affichage de cette ligne?
4. Si on décommente la ligne commentée **L3** du code ci-dessous alors cela produit-il une erreur? Si oui pourquoi? Si non quel sera l'affichage de cette ligne?

```
1 #include<iostream>
2 void f(char a) {std::cout<<"f1"<<std::endl;}
3 int f(long a ,char b){std::cout<<"f2"<<std::endl; return 1;}
4 char f(long a) {std::cout<<"f3"<<std::endl; return 'a';}
5 void f() {std::cout<<"f4"<<std::endl;}
6 //int f(short a) {std::cout<<"f5"<<std::endl;} // L1
7 int main(void){
8     f('c');
9     f(21);
10    f();
11    f(31, 'c');
12    //f('c',41); // L2
13    //f(4); // L3
14    return 0;
15 }
```

Exercice 3. *Membre*

La compilation C++ du code suivant produit une erreur. Indiquer à quelle ligne et la raison de l'erreur.

```
1 void methode1(){}
2 class A{
3     int a;
4     public:
5     void methode2(){ }
6     void methode3(){
7         methode1();
8         this->methode2();
9         methode2();
10    }
11 };
12 int main(){
13     A obj;
14     obj.methode2();
15     obj.methode1();
16     methode1();
17     return 0;
18 }
```

Exercice 4. *Construction ?*

La compilation C++ du code suivant produit une erreur. Indiquer à quelle ligne et la raison de l'erreur.

```
1 class A{
2     int a;
3
4     A(int a){
5         a=a;
6     }
7     A()    : A{3} {}
8 };
9 int main()
10 {
11     A obj{};
12     A obj2{3};
13     return 0;
14 }
```

Exercice 5. 3

Un objet de la classe **Chaine** doit contenir une liste de **char** (autrement appelée une chaîne de caractères). Terminer l'écriture de classe **Chaine** ci-dessous, il y a un exemple d'utilisation de la classe plus bas.

Pensez aux opérations usuelles (affectation, recopie, destruction).

```
class Chaine{
    int taille=0;
    char* t=NULL;
public:
    Chaine(const char* c) {
        for(taille=0;c[taille]!='\0';taille++){ // calcul de la longueur de la chaine
            t=(char*)malloc(sizeof(char)*taille);
            int i;
            for(i=0;i<taille;i++)
                t[i]=c[i];
        }
        /* reste à compléter ici */
    };
};
```

Exemple d'utilisation de la classe.

```
Chaine duplique(Chaine objet){
    objet=objet+objet;
    return objet;
}

int main(){
    Chaine obj{"joyeux "};
    Chaine obj2{"anniversaire. "};
    obj.get(1); // renvoie le second caractère ("o" donc)
    Chaine obj3=obj; // obj3 est une copie de obj
    obj3.get(0)==obj.get(0); // renvoie 1 (vrai)
    obj3.set(0,'J'); // modifie le premier caractère de obj3
    // obj3 contient maintenant "Joyeux "
    obj3.get(0)==obj.get(0); // renvoie 0 (faux)
    obj2=obj3+obj2; // + :concaténation
    // obj2: "Joyeux anniversaire. "

    obj=obj2;
    obj.get(0)==obj2.get(0); // renvoie 1 (vrai)
    obj.set(0,'a');
    obj.get(0)==obj2.get(0); // renvoie 0 (faux)
    obj2=obj; // Attention à ce cas !
    obj3=duplique(obj2);
    // ici obj3 doit contenir "Joyeux anniversaire. Joyeux anniversaire. "
    return 0; // toute la mémoire allouée doit être libérée
}
```