

EISE-4

Rapport projet C++ : Election, pièges à cons ?



Vote électronique

PANCHALINGAM

Niverthan

COUNDJIDAPADAME

Vikenesh

Polytech PARIS UPMC

I.	Introduction.....	2
	Contraintes et travail demandé	2
II.	Le déroulement du programme	2
III.	Diagramme de classe.....	4
IV.	Les classes et méthodes les plus importantes du programme	4
1.	La classe Personne.....	4
2.	Les classes Electeur et Candidat.....	5
3.	La méthode Influencer (Election *elect) de la classe Presse.....	6
4.	La classe Election	6
5.	Le main	7
V.	Conclusion	7

I. Introduction

Le projet proposé dans le cadre du module de C++ est le suivant : Election, pièges à cons ?

Il nous est donc demandé d'aborder le thème des élections en incluant un piège ou une escroquerie.

Contraintes et travail demandé

Les contraintes :

Le travail consiste à programmer en langage C++ le programme qui va gérer notre système :

- 8 classes
- 3 niveaux de hiérarchie ;
- 2 fonctions virtuelles différentes et utilisées à bon escient ;
- 2 surcharges d'opérateurs ;
- 2 conteneurs différents de la STL ;
- diagramme de classe UML complet ;
- commentaire du code ;
- pas d'erreurs avec Valgrind ;
- rendu par dépôt git, adresse à envoyer par mail avec dans le sujet le motif [EISE 4 C++ Projet] ;
- pas de méthodes/fonctions de plus de 30 lignes (hors commentaires, lignes vides et assert) ;
- utilisation d'un Makefile avec une règle "all" et une règle « clean ».

II. Le déroulement du programme

On compile le code grâce à la procédure make que l'on tape sur le terminal. On exécute le code en tapant «. ./<Nom_executable> ».

Une fois exécuté, le programme se lance et un menu apparaît montrant à l'utilisateur les candidats qui se présentent pour l'élection : On retrouve notamment des informations sur le nom, prénom, l'âge et le parti auquel les candidats appartiennent (dans notre cas, 3 candidats postulent pour devenir président).

```
Voici la liste des candidats pour cette election :  
Tour1  
-----  
Nom: Fillon  
Prenom: Francois  
Age: 62  
-----  
Nom: Valls  
Prenom: Manuel  
Age: 54  
-----  
Nom: LePen  
Prenom: Marine  
Age: 48  
-----  
Entrer votre nom: 
```

Figure 1 : Listes des candidats qui se présentent à l'élection

L'utilisateur a ensuite la possibilité en tant qu'électeur de pouvoir participer aux élections en votant.

Dans un premier temps des informations sur l'électeur sont demandées à être saisies au clavier. (On demande notamment son nom, prénom, âge...)

```
Entrer votre nom: Tintin
Entrer votre prenom: reporter
Entrer votre age: 15
Entrer votre id: 2235665
Quel est le nom de votre region de residence ?
IDF
```

Figure 2 : Information sur l'électeur

Ensuite l'utilisateur doit saisir sa région de résidence, afin de comptabiliser les voix par régions.

Enfin l'électeur peut voter pour le candidat qu'il désire en prenant soin d'écrire correctement le nom du candidat souhaité.

L'élection se fait en deux tours. L'utilisateur ayant la possibilité de voter, le reste des voix est générer aléatoirement.

```
Entrer votre nom: Tintin
Entrer votre prenom: Reporter
Entrer votre age: 25
Entrer votre id: 15663
Quel est le nom de votre region de residence ?
Aquitaine
--- Vote ---
Pour qui voulez-vous votez?
Valls
A voter!
```

Figure 3 : Vote de l'électeur

Finalement, un affichage final du résultat affiche le vainqueur des élections.

```
Le candidat élu est :
Nom: Valls
Prenom: Manuel
Age: 54
```

Figure 4 : Affichage du vainqueur

Ce système peut être considéré comme un simulateur de vote électronique comme l'on peut trouver dans certains pays.

III. Diagramme de classe

Le diagramme de classe ci-dessous résume la relation entre chaque classe (niveau de hiérarchie, association,...), on y retrouve également le nom des attributs et des méthodes de chaque classe.

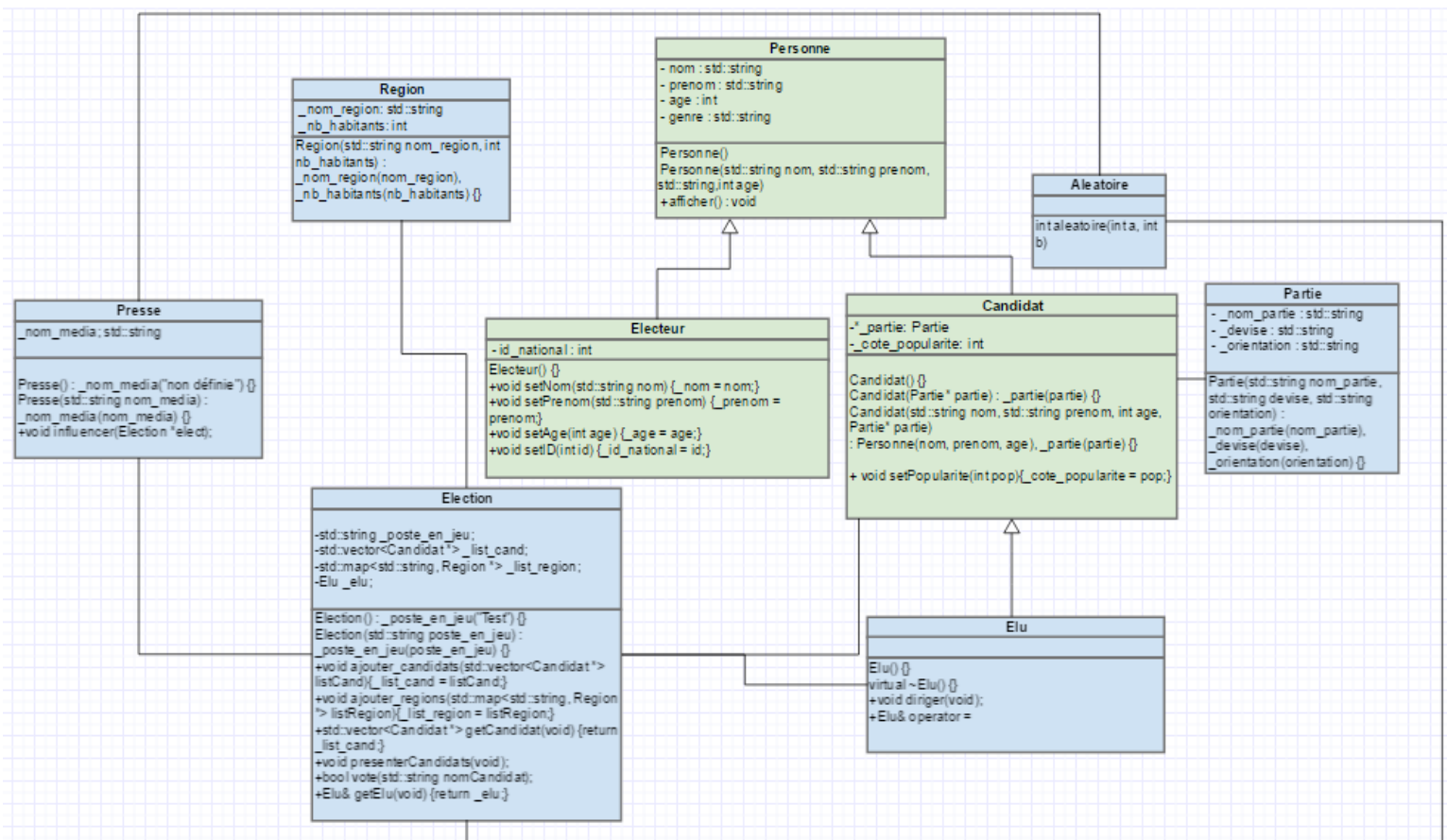


Figure 3 : Diagramme de classe

La classe mère du système est la classe Personne qui contient les informations concernant les candidats et électeurs. On note les trois niveaux de hiérarchie respectivement entre les classes Personne, Candidat et Elu. La classe Election est la classe qui fait le lien entre la plupart des classes, c'est notamment là que se déroule l'ajout des candidats et des électeurs.

IV. Les classes et méthodes les plus importantes du programme

1. La classe Personne

La classe `Personne` représente la classe mère du système. De cette classe descendent 2 classes : la classe `Electeur` et la classe `Candidat`.

Dans cette classe, on crée également la méthode `afficher()` qui sera utilisée pour les autres classes héritant de `Personne` en utilisant le principe d'héritage.

```
/* Méthode d'affichage de la classe Personne */
```

```
void Personne::afficher(void)
{
    std::cout << "nom:" << _nom << std::endl;
    std::cout << "prenom:" << _prenom << std::endl;
    std::cout << "age:" << _age << std::endl;
}
```

On crée également un constructeur qui instancie les attributs ainsi qu'un constructeur par défaut.

2. [Les classes `Electeur` et `Candidat`](#)

Ces deux classes héritent de la classe `Personne` et sont caractérisés notamment par la présence de liste dans lesquels sont rangées plusieurs candidats et électeurs.

```
/* Constructeurs et méthodes de la classe Electeur */
```

```
public:
    Electeur() {}
    void setNom(std::string nom) {_nom = nom;}
    void setPrenom(std::string prenom) {_prenom = prenom;}
    void setAge(int age) {_age = age;}
    void setID(int id) {_id_national = id;}
```

La classe `Electeur` contient les informations qui seront saisis par l'utilisateur au clavier. Cela fonctionne sur le même principe que le remplissage d'un formulaire ou de dossier administratif en ligne.

```
/* Constructeurs et méthodes de la classe Electeur */
```

```
class Candidat: public Personne
{
public:
    // Constructeur
    Candidat() {}
    Candidat(Partie* partie) : _partie(partie) {}
    Candidat(std::string nom, std::string prenom, int age, Partie* partie)
        : Personne(nom, prenom, age), _partie(partie) {}

    void setPopularite(int pop){_cote_popularite = pop;}
}
```

La classe `Candidat` est associée à un parti politique, on peut observer cette spécificité dans les deux constructeurs où un parti est passé en argument du constructeur.

3. La méthode Influencer (Election *elect) de la classe Presse

Nous avons également créé une classe Presse qui aurait le pouvoir d'influencer sur la cote de popularité d'un candidat. L'idée est donc de parcourir la liste de candidat créé à l'aide d'un itérateur et d'affecter aléatoirement une valeur de cote de popularité à chacun des candidats de la liste (On fixe une valeur maximale de popularité via un define).

```
/*Méthodes Influencer (Election *elect) */  
  
void Presse::influencer(Election *elect)  
{  
    std::vector<Candidat *> liste = elect->getCandidat();  
    std::vector<Candidat *>::iterator it;  
  
    for(it=liste.begin(); it != liste.end(); it++)  
    {  
        (*it)->setPopularite(aleatoire(0, POPULARITE_MAX));  
    }  
}
```

La fonction Aléatoire est utilisée ici pour affecter aléatoirement une valeur à la cote de popularité compris entre 0 et une valeur fixée par l'utilisateur dans le code.

La principale difficulté rencontrée dans cette classe a été le parcours et la modification d'un membre des éléments de la liste. Finalement en utilisant un container list de la STL, nous avons pu remédier à ce problème ainsi qu'en utilisant un itérateur comme vu en TP.

4. La classe Election

La classe Election constitue une des classes les plus importantes de notre système. En effet la classe Election contient les listes de candidats et des régions. Elle contient deux méthodes importantes que sont la présentation des candidats au lancement du programme et le vote automatique et aléatoire. En ce point, la classe Election contient le piège car bien que l'utilisateur ait la possibilité de voter, le vote se fait de manière aléatoire, et en conséquence un vainqueur aléatoire. Cependant, le nombre de candidats étant limité à 3, la chance de voir le candidat pour lequel on a voté, élu, reste importante.

```
/* Méthode d'affichage des candidats*/  
  
void Election::presenterCandidats(void)  
{  
    std::vector<Candidat *>::iterator cand;  
  
    for(cand = _list_cand.begin(); cand != _list_cand.end(); cand++)  
    {  
        std::cout << "-----" << std::endl;  
        (*cand)->afficher();  
        std::cout << "-----" << std::endl;  
    }
```

```
}  
}
```

Comme précédemment, on parcourt la liste des candidats et on affiche les informations les concernant.

5. [Le main](#)

Le main coordonne le menu de notre système de vote électronique. On y retrouve la plupart des affichages et des instanciations d'objets. Nous avons dans le cadre de la création des régions mis en place une map, qui nous permettait d'associer à une région, un nom, ainsi qu'un nombre d'habitants.

/ Création des régions */*

```
map<string, Region *> listeRegion;  
map<string, Region *>::iterator region;  
listeRegion["Ile_de_France"] = new Region("Ile_de_France", 257);  
listeRegion["Aquitaine"] = new Region("Aquitaine", 3564);  
listeRegion["Centre"] = new Region("Centre", 483);
```

Comme vue au paravent, le système de vote s'effectue sur 2 tours. Nous avons également mis en place des structures de vérifications lors de la saisies des noms de régions et des noms de candidats.

V. [Conclusion](#)

Ce projet nous a permis de d'approfondir et surtout d'appliquer nos connaissances en langage C++.

Cependant par manque de temps et de maîtrise, nous n'avons pas pu effectuer d'interface graphique qui aurait pourtant été adapté à notre système. La mise en place d'un diagramme de classe avant le codage a grandement facilité la compréhension du problème ainsi que les relations entre les classes. Ce projet a également été le moyen de mettre en application les éléments d'incompréhension pendant le cours et les TP.