Programmation orientée objet

LIV 2024-2025

Travaux pratiques 6

Site du cours : `https://defelice.up8.site/poo.html`
Les exercices marqués de (@) sont optionnels.
Un fichier écrit en langage C++ se termine conventionnellement par .cpp.
Une commande de compilation est `g++ fichier_source1.cpp fichier_source2.cpp fichier_source3.cpp`.
Voici des options de cette commande.

- `-o nom_sortie` pour donner un nom au fichier de sortie (par défaut `a.out`).
- `-Wall` pour demander au compilateur d'afficher plus d'avertissements (warnings)
- `-Wextra` pour demander au compilateur d'afficher plus d'avertissements (warnings)
- `-std=c++20` pour compiler selon la norme c++20
- `-g -fsanitize=address` pour que l'exécutable puisse detecter certaines erreurs de mémoire (executable plus lent, -g pour avoir plus de précision).
- `-g` pour ajouter des informations de débugage, utile avec debug (executable plus lent).

Exemple : `g++ -Wall -Wextra fichier1.cpp -o monprogramme`

Exercice 1. *Chaine*

Créer une classe `Chaine` qui hérite de la classe `std::string` et qui définit de nouvelles méthodes :

- `inverse(void)` qui modifie la chaine en la remplaçant par son miroir (`abcd` devient `dcba`)
- `cycle` qui modifie la chaine en décalant les lettres de la chaine. `cycle(-2)` sur `abcde` donne `cdeab` et `cycle()` doit décaler chaque lettre de 1 vers la droite `eabcd`.

```
Chaine ch1{"Bonjour"}; //
Chaine ch2{"eiluJ"}; // Julie à l'envers
ch2.inverse();
std::cout << ch1 << ch2 << std::endl; // Bonjour Julie
Chaine ch3{"iasEl"};
ch3.cycle(-3);
std::cout << ch1 << ch3 << std::endl; // Bonjour Elias
```

Exercice 2. *ChaineD*

Créer une classe `ChaineD` qui hérite de la classe `Chaine` et qui définit de nouvelles méthodes. Si `m` est un objet de type `ChaineD` alors :

- `m.shift(23)` ajoute la valeur 23 (modulo 256) à chaque octet.
- `hach` qui renvoie une signature numérique (un hachage de la chaîne sous la forme d'un `int`). Comme signature on prendra la (partie entière) de la somme des octets divisée par le nombre de lettres. (penser à la gestion des objets `ChaineD` const).
- `chiffre(int cle)` Qui chiffre la chaîne selon la méthode (très faible) suivante : On ajoute `cle*cle` (modulo 2^{16}) au premier octet (modulo 256), on ajoute `cle*cle*cle` (modulo 2^{16}) au second octet (modulo 256), on ajoute `cle*cle*cle*cle` (modulo 2^{16}) au troisième octet (modulo 256), et ainsi de suite jusqu'à la fin de la chaine.
- `dChiffre(int cle)` qui déchiffre la chaine chiffré selon le chiffrage précédent.

Exercice 3. *Espace de nom*

Écrire dans deux fichiers `chaine.hpp` et `chaine.cpp` les déclarations et définitions des deux classes précédentes.

La classe `ChaineD` doit être dans l'espace de nom `chaine` et la classe `Chaine` doit être dans l'espace de nom `chaine::chaineN`.