

Exercice sur la STL et nouveautés

- 1) Ecrire un code capable de créer des objets en fonction d'une chaîne de caractère

Préparer des prototypes d'objets dans une `std::map<const char*, MaClasse>`, puis lire la ligne de commande (la clé) et cloner l'objet trouvé en fonction de la clé.

- 2) Créer le code de la gestion de la bibliothèque

Une Bibliothèque contient des documents.

- Placer les documents dans un vector

```
orsys::Bibliotheque bib{ "orsys bib" };

bib.ajoute("A tour of C++", "Bjarne Stroustrup", 239);
bib.ajoute("Mastering the C++17 STL", "Arthur ODwyer", 361);
bib.ajoute("C++ High Performance", "Viktor Sehr", 352);
bib.ajoute("Bulletproof Android", "GodFrey Nolan", 207);

std::cout << bib << std::endl;
```

- Créer une map des documents pour lier un index au document

```
orsys::Bibliotheque bib{ "orsys bib" };
bib.ajoute("Guru","A tour of C++", "Bjarne Stroustrup", 239);
bib.ajoute("STL","Mastering the C++17 STL", "Arthur ODwyer", 361);
bib.ajoute("C++","C++ High Performance", "Viktor Sehr", 352);
bib.ajoute("Android", "Bulletproof Android", "GodFrey Nolan", 207);
```

Ressortir les informations :

```

auto cles = bib.get_cles();
for (auto k : cles) {
    std::cout << " cle " << k << std::endl;
}

// affiche tous les documents pour la clé C++
auto docs = bib.get_docs("C++");
std::cout << docs;

// affiche toutes les clés pour un document
auto cle = bib.get_cles(doc);
std::cout << cle << std::endl;

```

- Utiliser une multimap de documents pour avoir plusieurs documents pour une clé
- Créer une fonction retournant une `std::unordered_map` qui fait correspondre un document à tous ses index