

```
/*
-----
Laboratoire : 04
Fichier      : collection_g.h
Auteur(s)    : Thibaud Franchetti, Sacha Perdrizat
Date         : 08.04.2019

But          : Fournit une classe Collection générique

Remarque(s) : - La classe n'est pas garantie de fonctionner avec tout type
                de conteneurs, les paramètres génériques doivent être compatibles
                et ils doivent fournir les fonctions utilisées dans l'implémentation.

Compilateur  : GCC-g++ 7.3.0
               GCC-g++ 8.2.0
-----
*/

#ifndef COLLECTION_G_H
#define COLLECTION_G_H

#include <iostream>
#include <memory>           // std::allocator
#include <algorithm>
#include <sstream>
#include "exceptions.h"

template <typename T, template<typename, typename> class CONTENEUR>
class Collection;

template <typename T, template<typename, typename> class CONTENEUR>
std::ostream& operator<< (std::ostream& os, const Collection<T, CONTENEUR>& element);

template <typename T, template<typename, typename> class CONTENEUR>
class Collection {
    friend std::ostream& operator<< <>(std::ostream& os, const Collection& element);
public:
    Collection() = default;
    void ajouter(const T& element);
    size_t taille() const;
    T& get(size_t pos);
    bool contient(const T& element) const;
    void vider();
    template <typename UnaryOperator>
    void parcourir(const UnaryOperator& fonction);
private:
    CONTENEUR<T, std::allocator<T>> data;

    static std::string indiceNonValide(const char* nomFonction);
};

#include "collection_g_impl.h"

#endif
```