МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Дніпровський національний університет
залізничного транспорту імені академіка В. Лазаряна

Кафедра «Комп'ютерні інформаційні технології»

Лабораторна робота № 12

з дисципліни «Об'єктно-орієнтоване програмування»

на тему: «Асоціативні контейнери map, multimap»

Виконав:
студент гр.ПЗ1911
Сафонов Д. Є.
Прийняла:
Демидович І.М.

Дніпро, 2021

Тема. Асоціативні контейнери map, multimap

Завдання.
Написати програму яка демонструвала б роботу з map, multimap. Для демонстрації роботи використовувати операції додавання, видалення, перегляду і не менше трьох інших алгоритмів (методів) бібліотеки STL. При роботі з контейнерами використовувати ітератори.

Індивідуальне завдання. 6. Географічний довідник (країна і інформація про неї).

Текст програми.

“main.cpp”

```
#include "tests/gazetteer_tests.h"
#include "tests/gazetteer_multi_tests.h"
#include <iostream>

auto main() -> int {
	testGazetteerGood(); std::cout << std::endl;
	testGazetteerBad(); std::cout << std::endl;
	testGazetteerMultiGood(); std::cout << std::endl;
	testGazetteerMultiBad();
}
```

“gazetteer.h”

```
/// @file gazetteer.h

#ifndef GAZETTEER_H
#define GAZETTEER_H

#include <map>
#include <string>

/**
 * std::map wrapper to represent Gazetteer.
 */
class Gazetteer {
    std::map<std::string, std::string> dict;
public:
    /**
     * @brief Returns info about country.
     * @param[in] country Country to get info about
     * @return String with info about country
     * @details
     * Returns "No info about this country." if no info about this country.
     */
    [[nodiscard]] auto getInfo(const std::string& country) const -> std::string {
        auto val = this->dict.find(country);
        if (val == this->dict.end()) {
            return "No info about this country.";
        }
        return val->second;
    }

    /**
     * @brief Inserts info about country.
     * @param[in] country Country to insert info about
     * @param[in] info Info about country
     * @return true in case of success(see details)
     * @details
     * Inserts info about country if there is none, returns true.
     * If info already exists returns false.
     */
    auto insert(const std::string& country, const std::string info) -> bool {
        bool ret = !this->dict.contains(country);
        this->dict.insert({country, info});
        return ret;
    }
    /**
     * @brief Inserts info about country.
     * @param[in] country Country to insert info about
     * @param[in] info Info about country
     * @return true in case of insertion, false in case of replacement
     * Inserts info about country if there is none, returns true.
     * If info already exists, replaces it returns false.
     */
    auto insert_force(const std::string& country, const std::string info) -> bool {
        return this->dict.insert_or_assign(country, info).second;
    }
};

#endif
```

“gazetteer_multi.h”

```
/// @file gazetteer_multi.h

#ifndef GAZETTEER_MULTI_H
#define GAZETTEER_MULTI_H
#include <map>
#include <string>
#include <vector>
/**
 * Wrapper for std::multimap to represent Gazetteer.*/
class GazetteerMulti {
	std::multimap<std::string, std::string> multidict;
public:
	/**
	 * @brief Inserts new info about country.
	 * @param[in] country Country to insert info about
	 * @param[in] data Info about country
	 */
	void insertInfo(const std::string& country, const std::string data) {
		this->multidict.insert({country, data});
	}

	/**
	 * @brief Returns vector of inputs about country.
	 * @param[in] country Country to get info about
	 * @returns Vector of input about country(may be empty)
	 */
	auto getInfo(const std::string& country) -> std::vector<std::string> {
		auto [first, last] = multidict.equal_range(country);
		auto info = std::vector<std::string>();
		for (; first != last; first++) {
			info.push_back(first->second);
		}
		return info;
	}

	/**
	 * @brief Removes all inputs about country.
	 * @param[in] country Country to remove inputs about
	 */
	void removeCountry(const std::string& country) {
		multidict.erase(country);
	}
};
#endif
```

“tests/gazetteer_tests.h”

```
/// @file gazetteer_tests.h

#ifndef TESTS_GAZETTEER_TESTS_H
#define TESTS_GAZETTEER_TESTS_H

void testGazetteerGood();

void testGazetteerBad();

#endif
```

“tests/gazetteer_multi_tests.h”

```
/// @file gazetteer_multi_tests.h

#ifndef TESTS_GAZETTEER_MULTI_TESTS_H
#define TESTS_GAZETTEER_MULTI_TESTS_H

void testGazetteerMultiGood();

void testGazetteerMultiBad();

#endif
```

“tests/gazetteer_tests.cpp”

```
#include "gazetteer_tests.h"

#include "gazetteer.h"
#include <iostream>
#include <vector>

const std::string UA_LONG = "Ukraine is a country in Eastern Europe. It is bordered by Russia to the
east and north-east; Belarus to the north; Poland, Slovakia and Hungary to the west; and Romania,
Moldova, Crimea, and the Black Sea to the south. Including Crimea, Ukraine has an area of 603,628
km², and is the second-largest country in Europe after Russia.";
const std::string USA_LONG = "The United States—officially the United States of America,
commonly known as America—is a country primarily located in North America, consisting of 50
states, a federal district, five major self-governing territories, 326 reservations, and various
possessions.";
//cert-err58-cpp

void testGazetteerGood() {
	auto gazetteer = Gazetteer();

	gazetteer.insert("Ukraine", UA_LONG);
	gazetteer.insert("USA", USA_LONG);
	std::cout << "Gazetteer tests with correct input:" << std::endl;
	std::cout << "\t" << R"(insert("Ukraine", "...");)" << std::endl;
	std::cout << "\t" << R"(insert("USA", "...");)" << std::endl;

	auto countries = std::vector { "Ukraine", "USA" };
	auto print_countries = [&] () {
		for (const auto* country : countries) {
			std::cout << "\t\t" << country << ": " << gazetteer.getInfo(country) << std::endl;
		}
	};

	print_countries();
	auto ret = gazetteer.insert_force("Ukraine", "New info about Ukraine.");
	std::cout << "\t" << R"(insert_force("Ukraine", "New info about Ukraine.");)" << " -> " <<
std::boolalpha << ret << std::endl;
	print_countries();
}


void testGazetteerBad() {
	auto gazetteer = Gazetteer();
	gazetteer.insert("Ukraine", UA_LONG);
	gazetteer.insert("USA", USA_LONG);
```

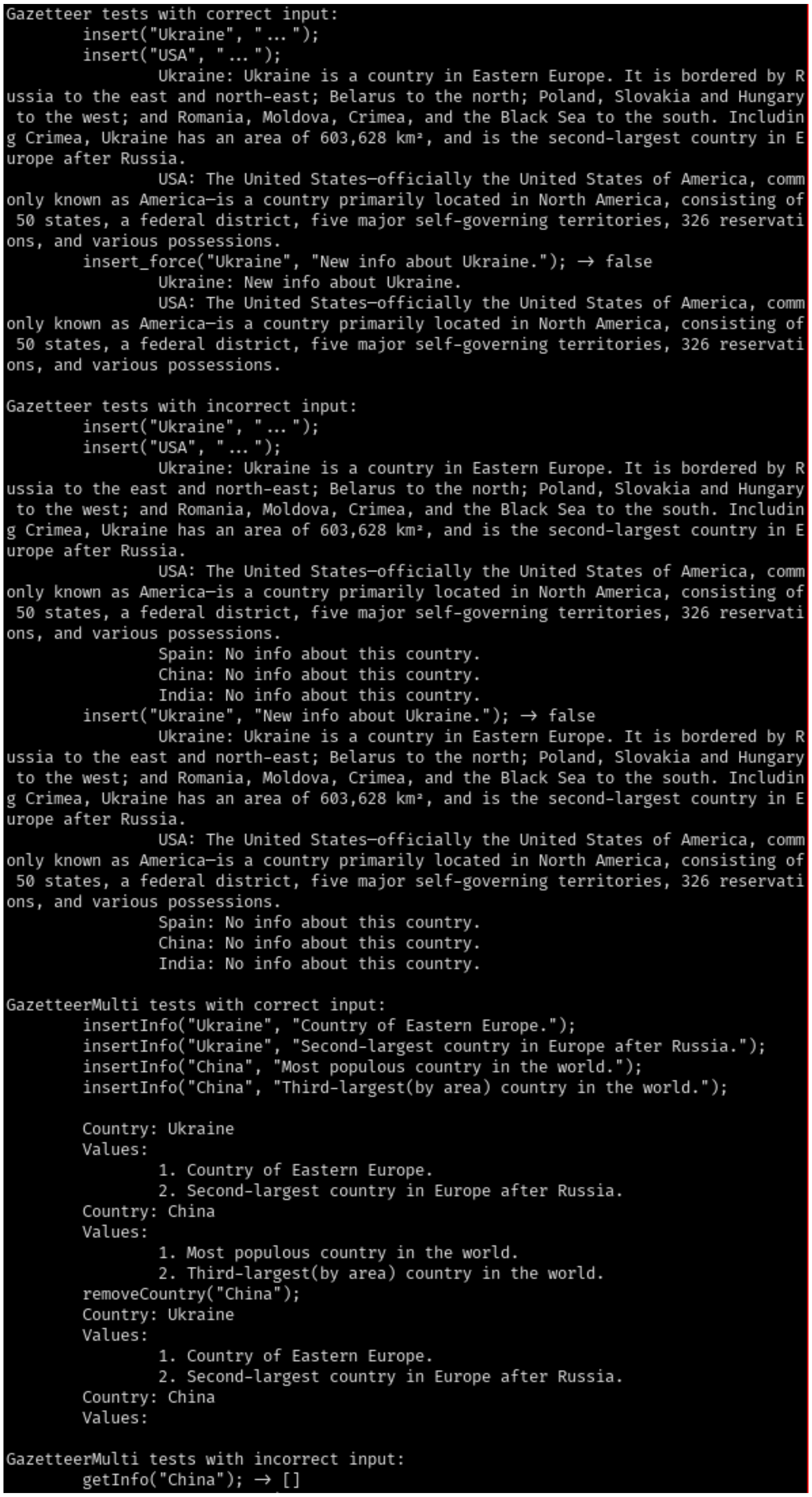
```
    auto countries = std::vector {
        "Ukraine",
        "USA",
        "Spain",
        "China",
        "India"
    };

    std::cout << "Gazetteer tests with incorrect input:" << std::endl;
    std::cout << "\t" << R"(insert("Ukraine", "...");)" << std::endl;
    std::cout << "\t" << R"(insert("USA", "...");)" << std::endl;
    for (const auto* country : countries) {
        std::cout << "\t\t" << country << ": " << gazetteer.getInfo(country) << std::endl;
    }

    auto ret = gazetteer.insert("Ukraine", "New info about Ukraine.");
    std::cout << "\t" << R"(insert("Ukraine", "New info about Ukraine.");)" << " -> " <<
std::boolalpha << ret << std::endl;
    for (const auto* country : countries) {
        std::cout << "\t\t" << country << ": " << gazetteer.getInfo(country) << std::endl;
    }
}
```

“tests/gazetteer_multi_tests.cpp”

```
#include "gazetteer_multi_tests.h"

#include "gazetteer_multi.h"
#include <iostream>

const std::string UA_SHORT1 = "Country of Eastern Europe.";
const std::string UA_SHORT2 = "Second-largest country in Europe after Russia.";

const std::string CHINA_SHORT1 = "Most populous country in the world.";
const std::string CHINA_SHORT2 = "Third-largest(by area) country in the world.";
//cert-err58-cpp

void testGazetteerMultiGood() {
    auto gazetteer = GazetteerMulti();
    auto pairs = std::vector {
        std::make_pair("Ukraine", &UA_SHORT1),
        std::make_pair("Ukraine", &UA_SHORT2),
        std::make_pair("China", &CHINA_SHORT1),
        std::make_pair("China", &CHINA_SHORT2),
    };

    std::cout << "GazetteerMulti tests with correct input:" << std::endl;
    for (auto& p : pairs) {
        gazetteer.insertInfo(p.first, *p.second);
        std::cout << "\t" << "insertInfo(\"" << p.first << "\", \"" << *p.second << "\");" << std::endl;
    }
    std::cout << std::endl;

    auto countries = std::vector { "Ukraine", "China" };
    auto print_countries = [&] () {
        for (const auto* country : countries) {
            auto values = gazetteer.getInfo(country);
            std::cout << "\tCountry: " << country << std::endl;
            std::cout << "\tValues:" << std::endl;
            for (size_t i = 0; i < values.size(); i++) {
                std::cout << "\t\t" << i + 1 << ". " << values[i] << std::endl;
            }
        }
    };
    print_countries();
    std::cout << "\t" << R"(removeCountry("China");)" << std::endl;
    gazetteer.removeCountry("China");
    print_countries();
}
```

```
void testGazetteerMultiBad() {
    auto gazetteer = GazetteerMulti();
    std::cout << "GazetteerMulti tests with incorrect input: " << std::endl;
    auto values = gazetteer.getInfo("China");
    std::cout << "\t" << R"(getInfo("China"); -> [)";
    for (auto value : values) {
        std::cout << value;// actually doesn't print anything, just to check that it in fact doesn't
    }
    std::cout << "]" << std::endl;
}
```

Приклад роботи програми.



```
Gazetteer tests with correct input:
        insert("Ukraine", "...");
        insert("USA", "...");
                Ukraine: Ukraine is a country in Eastern Europe. It is bordered by R
ussia to the east and north-east; Belarus to the north; Poland, Slovakia and Hungary
 to the west; and Romania, Moldova, Crimea, and the Black Sea to the south. Includin
g Crimea, Ukraine has an area of 603,628 km², and is the second-largest country in E
urope after Russia.
                USA: The United States—officially the United States of America, comm
only known as America—is a country primarily located in North America, consisting of
 50 states, a federal district, five major self-governing territories, 326 reservati
ons, and various possessions.
        insert_force("Ukraine", "New info about Ukraine."); → false
                Ukraine: New info about Ukraine.
                USA: The United States—officially the United States of America, comm
only known as America—is a country primarily located in North America, consisting of
 50 states, a federal district, five major self-governing territories, 326 reservati
ons, and various possessions.

Gazetteer tests with incorrect input:
        insert("Ukraine", "...");
        insert("USA", "...");
                Ukraine: Ukraine is a country in Eastern Europe. It is bordered by R
ussia to the east and north-east; Belarus to the north; Poland, Slovakia and Hungary
 to the west; and Romania, Moldova, Crimea, and the Black Sea to the south. Includin
g Crimea, Ukraine has an area of 603,628 km², and is the second-largest country in E
urope after Russia.
                USA: The United States—officially the United States of America, comm
only known as America—is a country primarily located in North America, consisting of
 50 states, a federal district, five major self-governing territories, 326 reservati
ons, and various possessions.
                Spain: No info about this country.
                China: No info about this country.
                India: No info about this country.
        insert("Ukraine", "New info about Ukraine."); → false
                Ukraine: Ukraine is a country in Eastern Europe. It is bordered by R
ussia to the east and north-east; Belarus to the north; Poland, Slovakia and Hungary
 to the west; and Romania, Moldova, Crimea, and the Black Sea to the south. Includin
g Crimea, Ukraine has an area of 603,628 km², and is the second-largest country in E
urope after Russia.
                USA: The United States—officially the United States of America, comm
only known as America—is a country primarily located in North America, consisting of
 50 states, a federal district, five major self-governing territories, 326 reservati
ons, and various possessions.
                Spain: No info about this country.
                China: No info about this country.
                India: No info about this country.

GazetteerMulti tests with correct input:
        insertInfo("Ukraine", "Country of Eastern Europe.");
        insertInfo("Ukraine", "Second-largest country in Europe after Russia.");
        insertInfo("China", "Most populous country in the world.");
        insertInfo("China", "Third-largest(by area) country in the world.");

        Country: Ukraine
        Values:
                1. Country of Eastern Europe.
                2. Second-largest country in Europe after Russia.
        Country: China
        Values:
                1. Most populous country in the world.
                2. Third-largest(by area) country in the world.
        removeCountry("China");
        Country: Ukraine
        Values:
                1. Country of Eastern Europe.
                2. Second-largest country in Europe after Russia.
        Country: China
        Values:

GazetteerMulti tests with incorrect input:
        getInfo("China"); → []
```

Рисунок 1

Висновки.

std::map<KeyType, ValueType> можна вважати обгорткою для std::vector<std::pair<KeyType, ValueType>>. A std::multimap<KeyType, ValueType> — для std::vector<std::pair<KeyType, std::vector<ValueType>>> . Тож використання цих класів можна замінити створенням класу який буде об’єднувати KeyType та ValueType або std::vector<ValueType> відповідно, або навіть просто мати поле із відповідною парою.

Але у випадках, коли дані які зберігаються у контейнерах локальні або контейнер константний, використання контейнерів набагато зручніше.