

**Міністерство освіти і науки України**  
**Львівський національний університет імені Івана Франка**  
**Факультет прикладної математики та інформатики**

**Кафедра програмування**

**Лабораторна робота №3**  
**Таблиця**  
з курсу “Алгоритми та структури даних”

Виконав:  
студент групи ПМІ-13  
Лук'янчук Денис  
Євгенійович

Львів – 2024

## Таблиця

**Таблиця** — це перелік, зведення статистичних даних або інших відомостей, розташованих у певному порядку за рядками та стовпчиками.

### Алгоритм виконання завдання:

1. Підключаємо необхідні бібліотеки, зокрема бібліотеку `<iomanip>`, яка використовується для форматованого виводу даних у консоль. Зокрема, функція `std::setw()` використовується для задання ширини поля виводу, що робить вигляд таблиці більш читабельним.
2. Визначаємо клас `Country`, який містить чотири публічні змінні: `name`, `capital`, `area`, і `population`. Клас також має конструктор, який ініціалізує об'єкт класу при його створенні, та метод `print()`, який виводить інформацію про країну.
3. Створюємо вектор `countries` типу `Country`. Додаємо об'єкти класу `Country` до вектора, ініціалізуючи їх даними про різні країни.
4. Використовуємо цикл `for`, в якому локальна змінна `i` (індекс) ініціалізується значенням 0. Цикл виконується, доки `i` менше розміру вектора `countries`. На кожній ітерації викликається метод `print()` для об'єкта `Country` за індексом `i`. Індекс `i` збільшується на 1 на кожній ітерації.
5. Використовуємо метод `print()` який виводить інформацію про країну в форматі, вказаному у коді, використовуючи функцію `std::setw()` для форматування виводу.

Використовуємо метод `print()` який виводить інформацію про країну в форматі, вказаному у коді, використовуючи функцію `std::setw()` для форматування виводу

### Приклад

Вхідні дані: Назви чотирьох країн, їхні площі, столиці і населення.

Country	Capital	Area	Population
Ukraine	Kyiv	603700	43700000
USA	Washington	9834000	332000000
United Kingdom	London	243600	67300000
Germany	Berlin	357600	83200000

## Приклад Unit-тесту без помилок

```
Country      Capital     Area    Population
Ukraine      Kyiv        603700   43700000
USA          Washington  9834000  332000000
United Kingdom London    243600   67300000
Germany      Berlin      357600   83200000
[=====] Running 4 tests from 1 test suite.
[-----] Global test environment set-up.
[-----] 4 tests from CountryTest
[ RUN   ] CountryTest.DisplayInfo_Ukraine
[ OK    ] CountryTest.DisplayInfo_Ukraine (1 ms)
[ RUN   ] CountryTest.DisplayInfo_USA
[ OK    ] CountryTest.DisplayInfo_USA (1 ms)
[ RUN   ] CountryTest.DisplayInfo_United_Kingdom
[ OK    ] CountryTest.DisplayInfo_United_Kingdom (2 ms)
[ RUN   ] CountryTest.DisplayInfo_Germany
[ OK    ] CountryTest.DisplayInfo_Germany (2 ms)
[-----] 4 tests from CountryTest (9 ms total)

[-----] Global test environment tear-down
[=====] 4 tests from 1 test suite ran. (11 ms total)
[ PASSED ] 4 tests.
```

## Приклад Unit-тесту з помилок

```
[=====] Running 4 tests from 1 test suite.
[-----] Global test environment set-up.
[-----] 4 tests from CountryTest
[ RUN   ] CountryTest.DisplayInfo_Ukraine
C:\Лычюїшсъш єр ёїїєъсєїш фрэш\Lab_3.1\Lab_3.1\Lab_3.1.cpp(36): error: Value of: output.find("603700") != string::npos
  Actual: false
Expected: true

[ FAILED  ] CountryTest.DisplayInfo_Ukraine (17 ms)
[ RUN   ] CountryTest.DisplayInfo_USA
[ OK    ] CountryTest.DisplayInfo_USA (5 ms)
[ RUN   ] CountryTest.DisplayInfo_United_Kingdom
[ OK    ] CountryTest.DisplayInfo_United_Kingdom (4 ms)
[ RUN   ] CountryTest.DisplayInfo_Germany
[ OK    ] CountryTest.DisplayInfo_Germany (6 ms)
[-----] 4 tests from CountryTest (34 ms total)

[-----] Global test environment tear-down
[=====] 4 tests from 1 test suite ran. (35 ms total)
[ PASSED ] 3 tests.
[ FAILED  ] 1 test, listed below:
[ FAILED  ] CountryTest.DisplayInfo_Ukraine

1 FAILED TEST
```

**Висновок:** У цьому програмному коді на мові C++, використовуючи клас та вектор, створюється проста таблиця інформації про країни. Кожен об'єкт класу 'Country' містить дані про назву країни, столицю, площу та населення. Для зручності та читабельності виведення використовуються бібліотеки <iostream> та <iomanip> для консольного вводу та форматованого виводу. Програма використовує вектор для динамічного зберігання об'єктів класу 'Country'. У циклі 'for' виводяться дані про країни у вигляді таблиці, де кожен рядок представляє окрему країну з її характеристиками. Додатково, використовуються бібліотеки '<vector>' та '<string>' для роботи з векторами та рядками відповідно. Цей код слугує прикладом простого використання класів та контейнера вектора для організації та виведення структурованих даних у C++.