

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 9
з навчальної дисципліни “Базові методології та технології програмування”
РЕАЛІЗАЦІЯ ПРОГРАМНИХ МОДУЛІВ РОЗГАЛУЖЕНИХ ТА
ІТЕРАЦІЙНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

ВИКОНАВ

студент академічної групи

КН-22 Ткаченко Роман

ПЕРЕВІРИВ

викладач кафедри кібербезпеки

та програмного забезпечення

_____ О.Г. Собінов

ЛАБОРАТОРНА РОБОТА №9

Тема: Реалізація програмних модулів розгалужених та ітераційних обчислювальних процесів.

Мета роботи: полягає у набутті ґрунтовних вмінь і практичних навичок реалізації технології модульного програмування, застосування операторів C/C++ арифметичних, логічних, побітових операцій, умови, циклів та вибору під час розроблення статичних бібліотек, заголовкових файлів та програмних засобів у кросплатформовому середовищі Code::Blocks.

Варіант №21

Завдання:

1. Реалізувати функції розв’язування задач 9.1–9.3 як складових статичної бібліотеки libModulesПрізвище.a (проект ModulesПрізвище, створений під час виконання лабораторної роботи №8).
2. Реалізувати програмне забезпечення розв’язування задачі 9.4 на основі функцій статичної бібліотеки libModulesПрізвище.a.

Хід Роботи

Завдання 9.1

— ЗАДАЧА 9.1 —

Вхід: статистичні дані чисельності населення області: загальна чисельність населення області станом на 1 січня ц.р. та кількість громадян, які станом на 1 число поточного місяця ц.р. знялись з реєстрації (вибули), зареєструвались на постійне місце проживання (прибули), народились, вмерли.

Вихід: чисельність населення області станом на 1 число поточного місяця та інформація про зменшення/збільшення кількості мешканців області у період з 1 січня ц.р. до 1 числа поточного місяця ц.р.

Строга постановка задачі

Вхідні данні: чисельність населення області, кількість смертельних випадків та кількість новонароджених, кількість іммігрантів та емігрантів (ціле число більше за нуль), станом на 1 число поточного місяця.

Вихідні данні: Чисельність населення з урахуванням всіх змінних та число зміни населення після обрахування

Лістинг модуля задачі 9.1

```
population_flow changes(int population, int birth, int death, int
emigration, int immigration){
    struct population_flow changes;
    changes.flow = birth - death + immigration - emigration;
    changes.population = population + changes.flow;
    return changes;
}
```

Завдання 9.2

— ЗАДАЧА 9.2 —

Вхід: зафіксовані в Кропивницькому показники швидкості переміщення повітряних мас (вітру, м/сек) щогодини.

Вихід: найменша швидкість вітру (в балах Бофорта), зафіксована продовж доби.

БАЛ БОФОРТА	ШВИДКІСТЬ ВІТРУ, м/сек	ДІЯ ВІТРУ
0	< 0.3	Відсутність вітру. Дим піднімається прямовисно. Листя дерев нерухомі.
1	0.3 – 1.5	Дим «пливе». Флюгер не обертається.
2	1.6 – 9.4	Рух повітря відчувається обличчям. Шелестить листя. Флюгери обертаються.
3	9.4 – 5.4	Тріпоче листя, хитаються дрібні гілки. Майорять прапори.
4	5.5 – 7.9	Хитаються тонкі гілки дерев. Вітер піднімає пил та шматки паперу.
5	8.0 – 10.7	Хитаються великі гілки. На воді з'являються хвилі.
6	10.8 – 13.8	Хитаються великі гілки
7	13.9 – 17.1	Хитаються невеликі стовбури дерев. На морі здіймаються хвилі, піняться.
8	17.2 – 20.7	Ламаються гілки дерев. і важко йти проти вітру.
9	20.8 – 24.4	Невеликі руйнування. Зриває покрівлі, руйнує димарі.
10	24.5 – 28.4	Значні руйнування. Дереву вибиваються з корінням
11	28.5 – 32.6	Великі руйнування
12	≥ 32.7	Призводить до спустошень

Строга постановка задачі

Вхідні данні: швидкість вітру щогодини на протязі останніх 24 годин (десяткове число більше нуля)

Вихідні данні: Сила вітру у Балах Бофорта найменшої швидкості вітру на протязі останніх 24 годин

Лістинг Модуля задачі 9.2

```
short int bal_boforta(float wind[24]){
    float Min = wind[0];
    for(int i = 1; i < 24; i++){
        if (wind[i] < Min){
            Min = wind[i];
        }
    }
}
```

```
    if (Min < 0.3){
        return 0;
    }
    else if (Min < 1.5){
        return 1;
    }
    else if (Min < 3.4){
        return 2;
    }
    else if (Min < 5.4){
        return 3;
    }
    else if (Min < 7.9){
        return 4;
    }
    else if (Min < 10.7){
        return 5;
    }
    else if (Min < 13.8){
        return 6;
    }
    else if (Min < 17.1){
        return 7;
    }
    else if (Min < 20.7){
        return 8;
    }
    else if (Min < 24.4){
        return 9;
    }
    else if (Min < 28.4){
        return 10;
    }
    else if (Min < 32.6){
        return 11;
    }
    else {
        return 12;
    }
}
```

Завдання 9.3

— ЗАДАЧА 9.3 —

Вхід: натуральне число N від 0 до 500700.

Вихід: якщо біт D_7 числа N рівний 0, кількість двійкових нулів у ньому, інакше — кількість двійкових нулів*.

*під час підрахунку кількості бінарних 0 або 1 рекомендовано використати тернарний оператор « ? : ».

Строга постановка задачі

Вхідні данні: Натуральне число в проміжку від 0 до 500700

Вихідні данні: якщо число по 7 індексу = 0 то кількість двійкових одиниць, якщо число = 1 то кількість двійкових одиниць

Лістинг модуля задачі 9.3

```
short int binary_func(unsigned int num){
    int zeros = 0;
    int ones = 0;
    bool return_zeros = false;
    for (int i = 0; num > 0; i++){
        if (i == 7){
            if (num%2 == 0){
                return_zeros = true;
            }
        }
        if (num%2 == 1){
            ones++;
        }
        else {
            zeros++;
        }
        num >>= 1;
    }
    if (return_zeros){
        return zeros;
    }
    else{
        return ones;
    }
}
```

TESTDRIVER

- 1) Підключаємо статичну бібліотеку «ModulesTkachenko.a» та заголовковий файл ModulesTkachenko.h.
- 2) Тестуємо очікуваний результат кожної функції (задач 9.1 – 9.3).

Лістинг тестового драйвера задач 9.1 – 9.3

```
#include <iostream>
#include <conio.h>
#include "ModulesTkachenko.h"

using namespace std;

int main()
{
    system("chcp 1251 & cls");
    int x, y, z;
    float wind_power[24];
    int population, death, birth, immigration, emigration;
    int numm;

    cout << "Hello! Program menu:" << endl
         << "d - function s_calculation" << endl
         << "j - function binary_func" << endl
         << "h - function bal_boforta" << endl
         << "g - function population_flow" << endl
         << "Press 'k', 'K', or 'Pe' to terminate the program." << endl;
    getchar();

    while(true) {
        cout << "Enter symbol (d/j/h/g/k/K/Pe): ";
        switch (getch()) {
            case 'd': cout << endl;
                     int x, y, z;
                     cout << "Enter x, y, z: ";
                     cin >> x >> y >> z;
                     cout << "Result: " << s_calculation(x, y, z) << endl;
                     break;
            case 'g': cout << endl;
                     cout << "Enter population of the region: ";
                     cin >> population;
```

```

        cout << "Enter number of birth on the region: ";
        cin >> birth;
        cout << "Enter number of death cases of the region: ";
        cin >> death;
        cout << "Enter number of emigration cases in the region:
";

        cin >> emigration;
        cout << "Enter number of immigration cases in the region:
";

        cin >> immigration;
        cout << "Current number of population in the region: " <<
changes(population, birth, death, emigration, immigration).population << endl;
        cout << "Current number of changes in the region: " <<
changes(population, birth, death, emigration, immigration).flow << endl;
        cout << endl;
        break;
    case 'h': cout << endl;
        for(int i = 0; i < 24; i++){
            cout << "Enter wind energy in " << i+1 << " hour: ";
            cin >> wind_power[i];
        }
        cout << "Beaufort score of the weakest wind power in 24
hours: " << bal_boforta(wind_power);
        cout << endl;
        break;
    case 'j': cout << endl;
        cout << "Enter number in range 0 - 500700: ";
        cin >> numm;
        cout << "The amount of binary zeros or ones: " <<
binary_func(numm) << endl;
        cout << endl;
        break;
    case 'k': return 0; break;
    case 'K': return 0; break;
    case 'Pe': return 0; break;
    default: cout << "\a" << endl << endl; continue;
}
}
}

```

Результат роботи TestDriver.exe

>>> Test of binary function:

Test #[1]: PASSED

Test #[2]: PASSED

Test #[3]: PASSED

Test #[4]: PASSED

Test #[5]: PASSED

>>> Test of Beaufort score function:

Test #[1]: PASSED

Test #[2]: PASSED

Test #[3]: PASSED

Test #[4]: PASSED

Test #[5]: PASSED

>>> Test of population_flow function:

Test #[1]: PASSED

Test #[2]: PASSED

Test #[3]: PASSED

Test #[4]: PASSED

Test #[5]: PASSED

Завдання 9.4

— ЗАДАЧА 9.4 —

За введенням користувачем символом “d” викликається `s_calculation()`, “g” – функція задачі 9.1, “h” – функція задачі 9.2, “j” – функція задачі 9.3; якщо користувач вводить інші символи, вони ігноруються, при чому видається звуковий сигнал про помилкове введення. Після цього, якщо користувач за запитом додатка вводить символ “k”, “K” або “к”, відбувається вихід з програми, інакше — виконання програми повторюється.

Алгоритмізація задачі 9.4

- 1) Підключаємо статичну бібліотеку «ModulesTkachenko.a» та заголовковий файл «ModulesTkachenko.h»
- 2) Вводимо символ
- 3) Якщо символ не співпадає з переліченими символами - повертаємося на крок 2

- 4) Якщо введений символ дорівнює «к», «К» або «k» - завершення програми
- 5) Якщо символ = d то
 - 5.1) Оголошуємо 3 цілочисельні змінні
 - 5.2) Вводимо 3 числа
 - 5.3) Виводимо результат функції s_calculation
- 6) Якщо символ = g
 - 6.1) Оголошуємо 5 цілочисельних змінних
 - 6.2) Вводимо 5 чисел
 - 6.3) Виводим результат функції population_flow_changes
- 7) Якщо символ = h
 - 7.1) Оголошуємо масив на 24 елементи
 - 7.2) Вводимо 24 елементи
 - 7.3) Виводимо результат функції bal_boforta
- 8) Якщо символ = j
 - 8.1) Оголошуємо цілочисельну змінну
 - 8.2) Вводимо число в діапазоні від 0 до 500700
 - 8.3) Виводимо результат функції binary_func
- 9) Користувач може вводити всі перелічені символи допоки не введе «к», «К» або «k» для завершення програми

Лістинг програми 9.4

```
#include <iostream>
#include <conio.h>
#include "ModulesTkachenko.h"

using namespace std;

int main()
{
    system("chcp 1251 & cls");
    int x, y, z;
    float wind_power[24];
    int population, death, birth, immigration, emigration;
    int numm;
```

```

cout << "Hello! Program menu:" << endl
    << "d - function s_calculation" << endl
    << "j - function binary_func" << endl
    << "h - function bal_boforta" << endl
    << "g - function population_flow" << endl
    << "Press 'k', 'K', or 'Pe' to terminate the program." << endl;
getchar();

while(true) {
    cout << "Enter symbol (d/j/h/g/k/K/Pe): ";
    switch (getch()) {
        case 'd': cout << endl;
            int x, y, z;
            cout << "Enter x, y, z: ";
            cin >> x >> y >> z;
            cout << "Result: " << s_calculation(x, y, z) << endl;
            break;
        case 'g': cout << endl;
            cout << "Enter population of the region: ";
            cin >> population;
            cout << "Enter number of birth on the region: ";
            cin >> birth;
            cout << "Enter number of death cases of the region: ";
            cin >> death;
            cout << "Enter number of emigration cases in the region:
";

            cin >> emigration;
            cout << "Enter number of immigration cases in the region:
";

            cin >> immigration;
            cout << "Current number of population in the region: " <<
changes(population, birth, death, emigration, immigration).population << endl;
            cout << "Current number of changes in the region: " <<
changes(population, birth, death, emigration, immigration).flow << endl;
            cout << endl;
            break;
        case 'h': cout << endl;
            for(int i = 0; i < 24; i++){
                cout << "Enter wind energy in " << i+1 << " hour: ";
                cin >> wind_power[i];
            }
    }
}

```

```

        cout << "Beaufort score of the weakest wind power in 24
hours: " << bal_boforta(wind_power);
        cout << endl;
        break;
    case 'j': cout << endl;
        cout << "Enter number in range 0 - 500700: ";
        cin >> numm;
        cout << "The amount of binary zeros or ones: " <<
binary_func(numm) << endl;
        cout << endl;
        break;
    case 'k': return 0; break;
    case 'K': return 0; break;
    case 'Pe': return 0; break;
    default: cout << "\a" << endl << endl; continue;
}
}
}

```

Висновок:

На основі виконаної лабораторної роботи можна зробити декілька обґрунтованих висновків.

По-перше, модульне програмування дозволяє ефективно розподіляти функціонал програми на окремі модулі, що сприяє збільшенню його читабельності та обслуговуваності. Використання заголовкових файлів та статичних бібліотек забезпечує зручний доступ до функцій з різних модулів.

По-друге, оператори C / C++ для роботи зі змінними та обчислень дозволяють ефективно вирішувати різноманітні завдання. Арифметичні та логічні операції забезпечують можливість здійснювати різноманітні обчислення, а оператори умови, циклів та вибору дозволяють забезпечити правильність виконання програми.

По-третє, Code::Blocks є зручним кросплатформовим середовищем для розробки програмних засобів. Воно має зручний інтерфейс, що спрощує роботу з файлами та проектами, а також вбудовані інструменти для налагодження та дебаггінгу програм.

В процесі виконання завдань лабораторної роботи виникло багато викликів та труднощів, але вирішення цих проблем дало мені цінний досвід. Також, робота в Code::Blocks дала мені можливість ознайомитись з цікавими можливостями цього середовища.

Щодо самого процесу виконання лабораторної роботи, варто відзначити, що він дозволяє досить глибоко зануритися в процес розробки програмного забезпечення та набути більш розгалужених навичок у програмуванні. Виконання завдань допомагає краще зрозуміти принципи роботи циклів, умовних операторів, розгалужень, та інших конструкцій, що допомагає у подальшій розробці проектів.

Отже, виконання лабораторних завдань допомагає набути вмінь і навичок у розробці програмного забезпечення, а також ознайомитися з кросплатформовим середовищем розробки. Варто зазначити, що під час виконання можуть виникнути складнощі, але вони допомагають краще зрозуміти принципи роботи програм та пошуку рішення в разі непередбачуваних ситуацій.

Додаток А

Назва тестового набору Test Suite Description	TS_9_1
Назва проекту / ПЗ Name of Project Software	population_flow_changes()
Рівень тестування Level of Testing	модульний

Автор тест-сюита Test Suite Author	Ткаченко Роман
Виконавець Implementer	Ткаченко Роман

Ід-р тест-кейса / Test Case ID	Дії (кроки) / Action (Test Steps)	Очікуваний результат / Expected Result	Результат тестування (пройшов/не вдалося/ заблокований) / Test Result (passed/failed/ blocked)
TCM_1	Вхід: 8543, 35, 4, 230, 452	Вихід: 8733, 190	PASSED
TCM_2	Вхід: 4550, 18, 5, 250, 300	Вихід: 4587, 37	PASSED
TCM_3	Вхід: 19500, 495, 98, 864, 957	Вихід: 19233, -267	PASSED
TCM_4	Вхід: 14596, 384, 98, 674, 547	Вихід: 14183, -413	PASSED
TCM_5	Вхід: 3495, 13, 6, 35, 56	Вихід: 3509, 14	PASSED

Додаток Б

Назва тестового набору Test Suite Description	TS_9_2
Назва проекту / ПЗ Name of Project Software	bal_boforta()
Рівень тестування Level of Testing	модульний

Автор тест-сьюта Test Suite Author	Ткаченко Роман
Виконавець Implementer	Ткаченко Роман

Ід-р тест-кейса / Test Case ID	Дії (кроки) / Action (Test Steps)	Очікуваний результат / Expected Result	Результат тестування (пройшов/не вдалося/ заблокований) / Test Result (passed/failed/ blocked)
TCM_1	Вхід: 11.5, 7.8, 6.3, 7.0, 20.3, 8.0, 7.2, 8.6, 12.2, 15.9, 21.7, 14.8, 18.6, 8.1, 24.8, 16.6, 18.2, 21.8, 15.7, 23.6, 19.1, 18.5, 15.3, 15.6	Вихід: 4	PASSED
TCM_2	Вхід: 0.2, 6.5, 8.2, 9.5, 11.2, 11.7, 11.9, 13.5, 14.8, 0.4, 18.1, 23.5, 24.2, 24.7, 25.9, 26.7, 28.1, 28.8, 29.3, 30.8, 31.9, 32.9, 36.1, 37.2	Вихід: 0	PASSED
TCM_3	Вхід: 22.5, 33.9, 21.4, 26.0, 28.5, 34.7, 38.0, 15.0, 31.2, 32.4, 29.8, 36.4, 35.8, 35.9, 38.9, 18.5, 20.2, 24.5, 12.8, 11.5, 20.1, 9.0, 32.0, 18.1	Вихід: 5	PASSED
TCM_4	Вхід: 15.2, 32.5, 9.8, 8.5, 12.2, 6.7, 7.3, 16.5, 9.3, 32.6, 33.9, 28.7, 9.8, 20.6, 27.9, 19.5, 23.6, 18.2, 21.3, 20.1, 0.9, 8.6, 27.0, 32.4	Вихід: 1	PASSED
TCM_5	Вхід: 30.0, 4.0, 1.0, 2.5, 8.5, 2.5, 21.5, 22.5, 30.0, 21.0, 26.5, 26.0, 6.5, 36.5, 0.5, 28.5, 14.0, 24.5, 13.5, 26.0, 37.2, 23.0, 0.3, 13.3	Вихід: 1	PASSED

Додаток В

Назва тестового набору Test Suite Descriptio	TS_9_3
--	--------

Назва проекту / ПЗ Name of Project Software	binary_func()
Рівень тестування Level of Testing	МОДУЛЬНИЙ
Автор тест-сюита Test Suite Author	Ткаченко Роман
Виконавець Implementer	Ткаченко Роман

Ід-р тест-кейса / Test Case ID	Дії (кроки) / Action (Test Steps)	Очікуваний результат / Expected Result	Результат тестування (пройшов/не вдалося/ заблокований) / Test Result (passed/failed/ blocked)
TCM_1	Вхід: 735249	Вихід: 12	PASSED
TCM_2	Вхід: 912315	Вихід: 15	PASSED
TCM_3	Вхід: 374679	Вихід: 13	PASSED
TCM_4	Вхід: 1265	Вихід: 6	PASSED
TCM_5	Вхід: 85429	Вихід: 10	PASSED

Додаток Г

Назва тестового набору Test Suite Description	TS_9_4
Назва проекту / ПЗ Name of Project Software	Tkachenko_task
Рівень тестування Level of Testing	системний / System Testing
Автор тест-сьюта Test Suite Author	Ткаченко Роман
Виконавець Implementer	Ткаченко Роман

Ід-р тест-кейса / Test Case ID	Дії (кроки) / Action (Test Steps)	Очікуваний результат / Expected Result	Результат тестування (пройшов/не вдалося/ заблокований) / Test Result (passed/failed/ blocked)
TS-1	1. Запустити програму	Hello! Program menu: d - function s_calculation j - function binary_function h - function bal_boforta g - function population_flow Press 'k', 'K', or 'к' to terminate the program.	PASSED
TS-2	1. Ввести "t" 2. Ввести d 3. Ввести 6 4. Ввести 8 5. Ввести 15	Hello! Program menu: d - function s_calculation j - function binary_function h - function bal_boforta g - function population_flow Press 'k', 'K', or 'к' to terminate the program. Enter symbol (d/j/h/g/k/K/к): *Звуковий сигнал*	PASSED

		Enter symbol (d/j/h/g/k/K/к): Enter x, y, z: 6 8 15 Result: -11.315 Enter symbol (d/j/h/g/k/K/к):	
TS- 3	1. Ввести "j" 2. Ввести 57395 3. Ввести "k"	Hello! Program menu: d - function s_calculation j - function binary_func h - function bal_boforta g - function population_flow Press 'k', 'K', or 'k' to terminate the program. Enter symbol (d/j/h/g/k/K/к): Enter number in range 0 - 500700: 57395 The amount of binary zeros or ones: 9 Enter symbol (d/j/h/g/k/K/к): *Закриття програми*	PASSED
TS- 4	1. Ввести "h" 2. Ввести 11.5, 7.8, 6.3, 7.0, 20.3, 8.0, 7.2, 8.6, 12.2, 15.9, 21.7, 14.8, 18.6, 8.1, 24.8, 16.6, 18.2, 21.8, 15.7, 23.6, 19.1, 18.5, 15.3, 15.6 3. Ввести "k"	Hello! Program menu: d - function s_calculation j - function binary_func h - function bal_boforta g - function population_flow Press 'k', 'K', or 'k' to terminate the program. Enter symbol (d/j/h/g/k/K/к): Enter wind energy in 1 hour: 11.5 Enter wind energy in 2 hour: 7.8 Enter wind energy in 3 hour: 6.3 Enter wind energy in 4 hour: 7.0 Enter wind energy in 5 hour: 20.3 Enter wind energy in 6 hour: 8.0 Enter wind energy in 7 hour: 7.2 Enter wind energy in 8 hour: 8.6 Enter wind energy in 9 hour: 12.2 Enter wind energy in 10 hour: 15.9 Enter wind energy in 11 hour: 21.7 Enter wind energy in 12 hour: 14.8	PASSED

		<p>Enter wind energy in 13 hour: 18.6</p> <p>Enter wind energy in 14 hour: 8.1</p> <p>Enter wind energy in 15 hour: 24.8</p> <p>Enter wind energy in 16 hour: 16.6</p> <p>Enter wind energy in 17 hour: 18.2</p> <p>Enter wind energy in 18 hour: 21.8</p> <p>Enter wind energy in 19 hour: 15.7</p> <p>Enter wind energy in 20 hour: 23.6</p> <p>Enter wind energy in 21 hour: 19.1</p> <p>Enter wind energy in 22 hour: 18.5</p> <p>Enter wind energy in 23 hour: 15.3</p> <p>Enter wind energy in 24 hour: 15.6</p> <p>Beaufort score of the weakest wind power in 24 hours: 4</p> <p>Enter symbol (d/j/h/g/k/K/к):</p> <p>*Закриття програми*</p>	
<p>TS-</p> <p>5</p>	<ol style="list-style-type: none"> 1. Ввести "г" 2. Ввести 3495 3. Ввести 13 4. Ввести 6 5. Ввести 35 6. Ввести 56 7. Ввести "к" 	<p>Hello! Program menu:</p> <p>d - function s_calculation</p> <p>j - function binary_func</p> <p>h - function bal_boforta</p> <p>g - function population_flow</p> <p>Press 'k', 'K', or 'Pe' to terminate the program.</p> <p>Enter symbol (d/j/h/g/k/K/к):</p> <p>Enter population of the region: 3495</p> <p>Enter number of birth on the region: 13</p> <p>Enter number of death cases of the region:</p> <p>6</p> <p>Enter number of emigration cases in the region: 35</p> <p>Enter number of immigration cases in the region: 56</p> <p>Current number of population in the region: 3509</p> <p>Current number of changes in the region: 14</p> <p>Enter symbol (d/j/h/g/k/K/к):</p> <p>*Закриття програми*</p>	<p>PASSED</p>

