

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Механіко-технологічний факультет

ЗВІТ
ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 9
з навчальної дисципліни
“Базові методології та технології програмування”

РЕАЛІЗАЦІЯ ПРОГРАМНИХ МОДУЛІВ РОЗГАЛУДЖЕННЯ
ТА ІТЕРАЦІЙНИХ ОБЧИСЛЮВАЛЬНИХ
ПРОЦЕСІВ

ЗАВДАННЯ ВИДАВ
доцент кафедри кібербезпеки
та програмного забезпечення
Доренський О. П.
<https://github.com/odorenskyi/>

ВИКОНАВ
студент академічної групи КБ-24
Жуковська Владислава

ПЕРЕВІРИВ
ст. викладач кафедри кібербезпеки
та програмного забезпечення
Коваленко Анастасія Сергіївна

ТЕМА: РЕАЛІЗАЦІЯ СТАТИСТИЧНИХ БІБЛОТЕК МОДУЛІВ ЛІНІЙНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

МЕТА: 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 22

— ЗАДАЧА 9.1 —

Вхід: набрані студентом бали: два рубіжні контролю до 30 балів кожен та семестровий екзамен до 40 балів. Вихід: семестрова оцінка (Rating).

Rating in scores	Rating a national scale	Rating scale ECTS	
		Rating	Explanation
90-100	Perfectly	A	excellent performance with small number errors
82-89	Good	B	above average some mistakes
75-81		C	overall correct fulfillment of certain significant number of errors
67-74	Satisfactorily	D	not bad, but with significant number of defects
60-66		E	satisfies minimum criteria
35-59	Unsatisfactorily	F	with the possibility of re-assembly
1-34		FX	with the obligatory repeated course

— ЗАДАЧА 9.2 —

Вхід: розмір взуття у см. Вихід: відповідник розміру в системі України.



Таблиця відповідності розмірів взуття в системах США, Європи, України, та в сантиметрах

Велико-британія	США		ЄС	см	Україна
	чоловічі	жіночі			
1	1,5	2,5	33	20,5	—
1,5	2	3	33 2/3	21	—
2	2,5	3,5	34 1/3	21,5	—
2,5	3	4	35	22	—
3	3,5	4,5	36	22,5	—
4	4,5	5,5	36 2/3	23	35
4,5	5	6	37 1/3	23,5	36
5	5,5	6,5	38	24	36,5
5,5	6	7	39	24,5	37
6	6,5	7,5	39 2/3	25	38
6,5	7	8	40 1/3	25,5	38/39
7,5	8	9	41	26	40,5

Велико-британія	США		ЄС	см	Україна
	чоловічі	жіночі			
8	8,5	9,5	42	26,5	41
8,5	9	10	42 2/3	27	41,5
9	9,5	10,5	43 1/3	27,5	42
9,5	10	11	44	28	42/43
10	10,5	11,5	45	28,5	43
11	11,5	12,5	45 2/3	29	44
11,5	12	13	46 1/3	29,5	45
12	12,5	13,5	47	30	46
12,5	13	14	48	30,5	47
13	13,5	14,5	48 2/3	31	47,5
14	14,5	15,5	49 1/3	31,5	48
14,5	15	16	50	32	48,5

— ЗАДАЧА 9.3 —

Вхід: натуральне число N від 0 до 50000000.

Вихід: якщо біт D_{10} числа N рівний 0, кількість двійкових нулів у ньому, інакше — кількість двійкових одиниць*.

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

— ЗАДАЧА 9.4 —

За введеним користувачем символом “u” викликається `s_calculation()`, “t” — функція задачі 9.1, “r” — функція задачі 9.2, “e” — 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 “w” або “i”, відбувається вихід з програми, інакше — виконання програми повторюється.



У випадку, якщо параметром `i` або результатом функції є дані нестандартного типу (наприклад, складового), то такий тип варто реалізувати у заголовковому файлі.

Аналіз і постановка задач 9.1, 9.2 та 9.3

Задача 9.1

Аналіз: Задача передбачає створення системи оцінювання, яка перетворює набрані студентом бали в семестрову оцінку за рейтинговою шкалою. Вхідними даними є результати двох рубіжних контролів (до 30 балів кожен) та семестрового екзамену (до 40 балів). На основі суми цих балів потрібно визначити підсумкову оцінку за національною шкалою та шкалою ECTS.

Постановка задачі:

- 1. Потрібно отримати від користувача бали за два рубіжні контролі та семестровий екзамен**
- 2. Підрахувати загальну суму балів**
- 3. На основі отриманої суми визначити рейтингову оцінку відповідно до таблиці:**
 - **90-100 балів: "Perfectly" (A)**
 - **82-89 балів: "Good" (B)**
 - **75-81 балів: "Good" (C)**
 - **67-74 балів: "Satisfactorily" (D)**
 - **60-66 балів: "Satisfactorily" (E)**
 - **35-59 балів: "Unsatisfactorily" (F)**
 - **1-34 балів: "Unsatisfactorily" (FX)**
- 4. Вивести результат у вигляді підсумкової оцінки (Rating)**

Задача 9.2

Аналіз: Задача полягає у створенні конвертера розмірів взуття. Користувач вводить розмір взуття в сантиметрах, а програма має визначити відповідний розмір в українській системі розмірів взуття, використовуючи надану таблицю відповідності.

Постановка задачі:

- 1. Отримати від користувача розмір взуття в сантиметрах**
- 2. Використовуючи таблицю відповідності, знайти найближчий український розмір взуття**
- 3. Вивести результат конвертації - український розмір взуття, що відповідає введеному значенню в сантиметрах**

Задача 9.3

Аналіз: Задача стосується обробки бінарного представлення числа. Для натурального числа N (від 0 до 500000000) потрібно визначити кількість двійкових нулів або одиниць у його бінарному представленні, залежно від значення біта D_{10} (десятого біта зправа).

Постановка задачі:

1. Отримати від користувача натуральне число N в діапазоні від 0 до 500000000
2. Перетворити число N у двійкове представлення
3. Перевірити значення десятого біта (D_{10}) числа N
4. Якщо D_{10} дорівнює 0, підрахувати кількість нулів у двійковому представленні
5. Якщо D_{10} не дорівнює 0, підрахувати кількість одиниць у двійковому представленні
6. Вивести результат підрахунку

Примітка: Рекомендовано використовувати тернарний оператор для визначення того, що саме рахувати (нулі чи одиниці).

Лістинг 9.1

```
char calculateRating(int test1, int test2, int exam) {  
  
    int total = test1 + test2 + exam;  
  
    if (total >= 90 && total <= 100) return 'A';  
    if (total >= 82) return 'B';  
    if (total >= 75) return 'C';  
    if (total >= 67) return 'D';  
    if (total >= 60) return 'E';  
    if (total >= 35) return 'F';  
    return 'FX';  
}
```

Лістинг 9.2

```
std::string convertShoeSize(double size_cm) {
    std::vector<std::pair<double, std::string>> sizeTable = {
        {20.5, "33"}, {21.0, "33 2/3"}, {21.5, "34"}, {22.0, "35"}, {22.5, "35 2/3"},
        {23.0, "36"}, {23.5, "36 2/3"}, {24.0, "37"}, {24.5, "38"}, {25.0, "38 2/3"},
        {25.5, "39"}, {26.0, "40"}, {26.5, "41"}, {27.0, "41.5"}, {27.5, "42"},
        {28.0, "42.5"}, {28.5, "43"}, {29.0, "44"}, {29.5, "45"}, {30.0, "46"},
        {30.5, "46.5"}, {31.0, "47"}, {31.5, "48"}, {32.0, "48.5"}
    };

    for (const auto& entry : sizeTable) {
        if (std::abs(size_cm - entry.first) < 0.01) {
            return entry.second;
        }
    }

    return "Unknown size";
}
```

Лістинг 9.3

```
int binaryBitCount(unsigned int N) {
    if ((N >> 10) & 1) == 0) {
        // Count number of 0 bits in binary
        int count = 0;
        for (int i = 0; i < 32; ++i)
            if ((N >> i) & 1) == 0) count++;
        return count;
    } else {
        // Count number of 1 bits
        return std::bitset<32>(N).count();
    }
}
```

Лістинг TestDriver

```
#include "ModulesZhukovsky.h"
#include <iostream>
#include <cstdlib>
#include <cmath>

int main() {
    char choice;
    bool continueProgram = true;

    while (continueProgram) {
        std::cout << "Введіть ваш вибір (u, t, r, e): ";
        std::cin >> choice;

        switch (choice) {
            case 'u': {
                double x, y, z;
                std::cout << "Введіть x, y, z для s_calculation: ";
                std::cin >> x >> y >> z;
                double result = s_calculation(x, y, z);
                std::cout << "Результат: " << result << std::endl;
                break;
            }
            case 't': {
                int test1, test2, exam;
                std::cout << "Введіть бали за тест1, тест2 та іспит: ";
                std::cin >> test1 >> test2 >> exam;
                char grade = calculateRating(test1, test2, exam);
                std::cout << "Оцінка: " << grade << std::endl;
                break;
            }
            case 'r': {
                double size_cm;
                std::cout << "Введіть розмір взуття в сантиметрах: ";
                std::cin >> size_cm;
                std::string shoeSize = convertShoeSize(size_cm);
                std::cout << "Розмір взуття: " << shoeSize << std::endl;
                break;
            }
            case 'e': {
                unsigned int N;
                std::cout << "Введіть ціле беззнакове число: ";
                std::cin >> N;

                if (N > 500000000) {
```

```

        std::cout << "Некоректне число. Повинно бути менше або рівне
500000000.\n";
    } else {
        // Обчислення кількості бітів та виведення результату
        std::cout << "Кількість бітів (в залежності від D10): " <<
binaryBitCount(N) << "\n";
    }
    break;
}
default:
    // Відтворення звукового сигналу про помилкове введення
    std::cout << "\a" << "Неправильне введення!" << std::endl;
    break;
}

char addChoice;
std::cout << "Бажаєте продовжити? Введіть 'w' або 'i' для виходу, будь-який
інший символ для продовження: ";
std::cin >> addChoice;

if (addChoice == 'w' || addChoice == 'i') {
    continueProgram = false;
}
}

return 0;
}

```

Аналіз і постановка задачі 9.4

Задача 9.4

Аналіз: Задача передбачає розробку інтерфейсу користувача для виклику функцій інших задач (9.1, 9.2, 9.3) за допомогою символьних команд. Система повинна реагувати на введені користувачем символи та викликати відповідні функції або сигналізувати про помилкове введення. Також передбачений вихід з програми за певними командами.

Постановка задачі:

1. Створити інтерфейс, що очікує введення користувачем символів:

- "u" - викликає функцію `s_calculation()`, яка відповідає задачі 9.1 (оцінювання студента)
- "r" - викликає функцію задачі 9.2 (конвертер розмірів взуття)

- "e" - викликає функцію задачі 9.3 (підрахунок бінарних нулів/одиниць)
 - інші символи мають ігноруватися з виведенням звукового сигналу про помилкове введення
2. Після виконання однієї з функцій, програма має перевірити наступний символ від користувача:
 - якщо введено "w" або "i", програма завершує роботу
 - якщо введено інший символ, програма продовжує виконання з початку циклу
 3. В разі, якщо параметри або результати функцій містять дані нестандартного типу (наприклад, складові), такі типи даних мають бути реалізовані в окремому заголовковому файлі.
 4. Забезпечити обробку помилок та коректне завершення програми.

Лістинг програми Zhukovska_task

```
#include "ModulesZhukovsky.h"
#include <windows.h>
#include <iostream>
#include <cstdlib>
#include <cmath>

int main() {
    SetConsoleOutputCP(65001);
    SetConsoleCP(65001);
    developer_info();
    char choice;
    bool continueProgram = true;

    while (continueProgram) {
        std::cout << "Введіть ваш вибір (u, t, r, e): ";
        std::cin >> choice;

        switch (choice) {
            case 'u': {
                double x, y, z;
                std::cout << "Введіть x, y, z для s_calculation: ";
                std::cin >> x >> y >> z;
                double result = s_calculation(x, y, z);
                std::cout << "Результат: " << result << std::endl;
```



```

        break;
    }
    case 't': {
        int test1, test2, exam;
        std::cout << "Введіть бали за тест1, тест2 та іспит: ";
        std::cin >> test1 >> test2 >> exam;
        char grade = calculateRating(test1, test2, exam);
        std::cout << "Оцінка: " << grade << std::endl;
        break;
    }
    case 'r': {
        double size_cm;
        std::cout << "Введіть розмір взуття в сантиметрах: ";
        std::cin >> size_cm;
        std::string shoeSize = convertShoeSize(size_cm);
        std::cout << "Розмір взуття: " << shoeSize << std::endl;
        break;
    }
    case 'e': {
        unsigned int N;
        std::cout << "Введіть ціле беззнакове число: ";
        std::cin >> N;
        int bitCount = binaryBitCount(N);
        std::cout << "Кількість бітів: " << bitCount << std::endl;
        break;
    }
    default:
        // Відтворення звукового сигналу про помилкове введення
        std::cout << "\a" << "Неправильне введення!" << std::endl;
        break;
}

char addChoice;
std::cout << "Бажаєте продовжити? Введіть 'w' або 'i' для виходу, будь-який
інший символ для продовження: ";
std::cin >> addChoice;

if (addChoice == 'w' || addChoice == 'i') {
    continueProgram = false;
}

}

return 0;
}

```

Висновок до лабораторної роботи

У ході виконання лабораторної роботи №9 на тему "Реалізація програмних модулів розгалужених та ітераційних обчислювальних процесів" було досягнуто значних результатів у контексті набуття ґрунтовних знань та практичних навичок модульного програмування. Робота була зосереджена на застосуванні операторів C/C++ для реалізації складних обчислювальних алгоритмів у середовищі Code::Blocks.

Першим етапом роботи стало детальне вивчення теоретичних основ модульного програмування, що дозволило зрозуміти принципи створення статичних бібліотек та програмних засобів для вирішення ітераційних та розгалужених обчислювальних задач. Було розглянуто особливості використання арифметичних, логічних та побітових операцій, а також структур умовного переходу та циклів при розробці програмного забезпечення.

У практичній частині роботи було реалізовано функції для розв'язання задач 9.1-9.3 як складових статичної бібліотеки libModules на основі проекту, створеного в межах попередньої лабораторної роботи. Під час реалізації було застосовано оптимальні підходи для обробки даних, що забезпечило ефективне виконання алгоритмів з мінімальним використанням обчислювальних ресурсів. Також було розроблено програмне забезпечення для розв'язання задачі 9.4 з використанням функцій бібліотеки libModules, що продемонструвало можливість повторного використання коду та переваги модульного підходу.

Особлива увага приділялася правильній організації коду з використанням заголовкових файлів, що дозволило ефективно розділити інтерфейс та реалізацію програмних компонентів. Це суттєво підвищило читабельність коду та спростило його подальшу підтримку. Для забезпечення контролю версій та ефективної співпраці було використано Git-репозиторій, що дозволило зберігати історію змін та організовувати роботу над проектом.

Під час виконання роботи виникли певні труднощі, пов'язані з налаштуванням середовища розробки та забезпеченням сумісності між різними компонентами програми. Зокрема, довелося вирішувати проблеми з підключенням статичних бібліотек та правильним визначенням шляхів до заголовкових файлів. Проте,

завдяки систематичному підходу до тестування та налагодження коду, ці проблеми були успішно вирішені.

Важливим аспектом роботи стало використання файлу-шаблону тестового набору для перевірки правильності роботи реалізованих функцій, що дозволило виявити та виправити помилки на ранніх етапах розробки. Тестування програми проводилося на різних наборах вхідних даних, що забезпечило перевірку роботи алгоритмів у різних умовах та підтвердило їх надійність.

У результаті виконання лабораторної роботи було набуто цінний досвід розробки модульних програм, що може бути використаний у майбутніх проектах. Зокрема, було закріплено навички використання операторів C/C++, роботи з умовними та циклічними конструкціями, створення та використання статичних бібліотек, а також організації коду з використанням заголовкових файлів. Розроблене програмне забезпечення відповідає всім вимогам завдання та демонструє ефективність обраних алгоритмічних рішень.

Таким чином, проведена лабораторна робота сприяла формуванню комплексного розуміння принципів модульного програмування та розвитку практичних навичок розробки програмного забезпечення, що є важливою складовою професійної компетентності фахівця в галузі інформаційних технологій. Отримані знання та навички можуть бути використані для розробки більш складних програмних систем, що вимагають ефективної організації коду та оптимального використання обчислювальних ресурсів.