

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 РОЗГАЛУЖЕНИХ ТА ІТЕРАЦІЙНИХ
ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

ВИКОНАВ
студент академічної групи
КН 22

_____ Червоний Є. В.

ПЕРЕВІРИВ
викладач кафедри кібербезпеки
та програмного забезпечення

_____ Олександр СОБІНОВ

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.

Завдання до лабораторної роботи

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

Варіант 29

ВАРІАНТ 29

— ЗАДАЧА 9.1 —

Вхід: бал Бофорта. Вихід: сила вітру, км/год.

БАЛ БОФОРТА	ШВИДКІСТЬ ВІТРУ, м/сек	ДІЯ ВІТРУ
0	< 0.3	Відсутність вітру. Дим піднімається вертикально. Листя дерев нерухомі.
1	0.3 – 1.5	Дим кливен. Флюгер не обертається.
2	1.6 – 3.4	Рух повітря відчувається обличчям. Шелестить листя. Флюгеро обертається.
3	3.5 – 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	Лавуються гілки дерев. І ваюю йти проти вітру.
9	20.8 – 24.4	Невеликі руйнування. Зриває покрівлі, руйнує димарі.
10	24.5 – 28.4	Значне руйнування. Деревя вириваються з корінням
11	28.5 – 32.6	Великі руйнування.
12	≥ 32.7	Приводить до спустошень

— ЗАДАЧА 9.2 —

Вхід: розмір в системі України. Вихід: розмір взуття в см.

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

Велико-британія	США		ЕС	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	7.5	8	41	26	40.5

Велико-британія	США		ЕС	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 1000 до 10000.

Вихід: якщо біт D_{15} 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 введенням символом “?” викликається `s_calculation()`, “5” – функція задачі 9.1, “4” – функція задачі 9.2, “3” – функція задачі 9.3; якщо вводяться інші символи, вони ігноруються та видається звуковий сигнал про помилкове введення. Після цього, якщо користувач вводить “9” або “0”, 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.

Хід роботи

Завдання 1

Завдання 9.1

Здійснено аналіз і постановку задачі 9.1. Виконано аналіз вимог, проектування архітектури, детальне проектування програмного модуля розв’язування задачі 9.1.

Алгоритмізувавши задачу отримаємо:

Вхідні данні: `p`.

Вихідні данні: `result`.

Обмеження, допущення:

`p` — Ціле число.

`result` — Рядок.

Лістинг задачі 9.1

```
string task9_1(int point)
{
    if (point >= 0 && point <=12)
    {
        switch (point)
        {
            case 0:
                return "<1.08";
            case 1:
                return "1.08-5.4";
            case 2:
                return "5.76-33.84";
            case 3:
                return "33.84-19.44";
            case 4:
                return "19.8-28.4";
            case 5:
                return "28.8-38.52";
            case 6:
                return "38.88-49.68 ";
            case 7:
                return "50.04-61.56";
            case 8:
                return "61.92-74.52";
            case 9:
```

```

        return "74.88-87.84";
    case 10:
        return "88.2-102.24";
    case 11:
        return "102.6-117.36";
    case 12:
        return ">=117.72";
    }
}
}

```

Завдання 9.2

Здійснено аналіз і постановку задачі 9.2. Виконано аналіз вимог, проектування архітектури, детальне проектування програмного модуля розв’язування задачі 9.2.

Алгоритмізувавши задачу отримаємо:

Вхідні данні: footsize.

Вихідні данні: result.

Обмеження, допущення:

filesize, result — Дійсні числа.

Лістинг задачі 9.2

```

float task9_2(float footsize)
{
    if(footsize == 35)
        return 23;
    if(footsize == 36)
        return 23.5;
    if(footsize == 36.5)
        return 24;
    if (footsize == 37)
        return 24.5;
    if (footsize == 38)
        return 25;
    if (footsize == 39)
        return 25.5;
    if (footsize == 40.5)
        return 26;
    if (footsize == 41)
        return 26.5;
    if (footsize == 41.5)
        return 27;
}

```

```

if (footsize == 42)
    return 27.5;
if (footsize == 43)
    return 28.5;
if (footsize == 44)
    return 29;
if (footsize == 45)
    return 29.5;
if (footsize == 46)
    return 30;
if (footsize == 47)
    return 30.5;
if (footsize == 47.5)
    return 31;
if (footsize == 48)
    return 31.5;
if (footsize == 48.5)
    return 32;
}

```

Завдання 9.3

Здійснено аналіз і постановку задачі 9.3. Виконано аналіз вимог, проектування архітектури, детальне проектування програмного модуля розв’язування задачі 9.3.

Алгоритмізувавши задачу отримаємо:

Вхідні данні: number.

Вихідні данні: count.

Обмеження, допущення:

number, count — Цілі числа.

Лістинг задачі 9.3:

```

int task9_3(int number)
{
    if (number >=1000 && number <=10000)
    {
        int bit = (number >> 14) & 1;
        int count = 0;
        if (bit == 0)
        {
            while (number != 0)
            {
                count += (number & 1) == 0 ? 1 : 0;
            }
        }
    }
}

```

```

        number >>= 1;
    }
    return count;
}
else
{
    while (number != 0)
    {
        count += (number & 1) == 1 ? 1 : 0;
        number >>= 1;
    }
    return count;
}
}
}

```

Виконано unit-тестування задач 9.1-9.3 за допомогою створеного TestDriver

Результат тестування:

Unit testing for task 9.1

Enter Bophor point: 5

Result: 28.8-38.52

Test#1 Passed

Enter Bophor point: 8

Result: 61.92-74.52

Test#2 Passed

Enter Bophor point: 12

Result: >=117.72

Test#3 Passed

Enter Bophor point: 10

Result: 88.2-102.24

Test#4 Passed

Unit testing for task 9.2

Enter Ukraine foot size: 35

Result: 23

Test#1 Passed

Enter Ukraine foot size: 40.5

Result: 26

Test#2 Passed

Enter Ukraine foot size: 42

Result: 27.5

Test#3 Passed

Enter Ukraine foot size: 48.5

```
Result: 32
Test#4 Passed
Unit testing for task 9.3
Enter number: 1000
Result: 4
Test#1 Passed
Enter number: 5000
Result: 8
Test#2 Passed
Enter number: 3568
Result: 5
Test#3 Passed
Enter number: 9985
Result: 9
Test#4 Passed
```

Завдання 2

Виконано аналіз вимог до ПЗ, проектування архітектури, детальне проектування програмного забезпечення розв'язування задачі 9.4

Алгоритмізувавши задачу отримаємо:

Вхідні данні: action, x, y, z, p, footsize, number.

Вихідні данні: результати виконання модулів: s_calculation(x, y, z), task9_1(p), task9_2(footsize), task9_3(number), звуковий сигнал при неправильному вводі.

Обмеження, допущення:

action, p, number — Цілі числа.

x, y, z, footsize — Дійсні числа.

Лістинг задачі 9.4:

```
#include <iostream>
#include "ModulesChervonyi.h"
using namespace std;
int main()
{
    while (true)
    {
        int action;
        cout<< "Enter action: "; cin >> action;
        switch(action)
        {
```

```

        case 7:
            float x,y,z;
            cout << "x: "; cin >> x;
            cout << "y: "; cin >> y;
            cout << "z: "; cin >> z;
            cout << "S= " << s_calculation(x, y, z) << endl;
            break;
        case 5:
            int p;
            cout << "Enter Bophor point: "; cin >> p;
            cout << task9_1(p) << endl;
            break;
        case 4:
            float footsize;
            cout << "Enter Ukraine foot size: "; cin >> footsize;
            cout << task9_2(footsize) << endl;
            break;
        case 3:
            int number;
            cout << "Enter number (1000 - 10000): "; cin >> number;
            cout << task9_3(number) << endl;
            break;
        default:
            if (action == 9 || action == 0)
                exit(0);
            else
                cout << "\a";
    }
}
}

```

Виконано системне тестування за розробленим TestSuite9.4.

Результат тестування:

```

Enter action: 7
x: 1
y: 0.2
z: 0.5
S= 1.54051
Enter action: 5
Enter Bophor point: 6
38.88-49.68
Enter action: 4
Enter Ukraine foot size: 42
27.5

```


Enter action: 3
Enter number (1000 - 10000): 5555
5
Enter action: 675
Enter action: 0

Process returned 0 (0x0)

Додатки:

Додаток 1-TestSuite9.1

Назва тестового набору	TestSuite9.1
Рівень тестування	Модульний(Unit-Testing)
Автор	Червоний Єгор Владиславович
Виконавець	Червоний Єгор Владиславович

ID	Введені дані	Виведені дані	Результат
1	point = 5	power = 28.8-38.52	PASSED
2	point = 8	power = 61.92-74.52	PASSED
3	point = 12	power = >=117.72	PASSED
4	point = 10	power = 88.2-102.24	PASSED

Додаток 2-TestSuite9.2

Назва тестового набору	TestSuite9.2
Рівень тестування	Модульний(Unit-Testing)
Автор	Червоний Єгор Владиславович
Виконавець	Червоний Єгор Владиславович

ID	Введені дані	Виведені дані	Результат
1	Ukraine size = 35	size = 23	PASSED
2	Ukraine size = 40,5	size = 26	PASSED
3	Ukraine size = 42	size = 27,5	PASSED
4	Ukraine size = 48,5	size = 32	PASSED

Додаток 3 -TestSuite9.3

Назва тестового набору	TestSuite9.3
Рівень тестування	Модульний(Unit-Testing)
Автор	Червоний Єгор Владиславович
Виконавець	Червоний Єгор Владиславович

ID	Введені дані	Виведені дані	Результат
1	number = 1000	count = 4	PASSED
2	number = 5000	count = 8	PASSED
3	number = 3568	count = 5	PASSED
4	number = 9985	count = 9	PASSED

Додаток 4 – TestSuite9.4

Назва тестового набору	TestSuite9.4
Рівень тестування	Системний
Автор	Червоний Єгор Владиславович
Виконавець	Червоний Єгор Владиславович

ID	Дії	Очікуваний результат	Результат
1	Запустити програму	Enter action:	PASSED
2	Ввести 7	x:	PASSED
3	Ввести значення x: 1	y:	PASSED
4	Ввести значення y: 0,2	z:	PASSED
5	Ввести значення z: 0,5	S= 1.54051 Enter action:	PASSED
6	Ввести 5	Enter Bophor point:	PASSED
7	Ввести значення 6	38.88-49.68 Enter action:	PASSED
8	Ввести 4	Enter Ukraine foot size:	PASSED
9	Ввести значення 42	27.5 Enter action:	PASSED
10	Ввести 3	Enter number (1000 - 10000):	PASSED
11	Ввести значення 5555	5	PASSED

		Enter action:	
12	Ввести значення 675	Програма відтворює сигнал про помилкове введення та виводить Enter action:	PASSED
13	Ввести значення 0	Програма завершить роботу	PASSED

Додаток 4 - Лістинг статичної бібліотеки ModulesChervonyi.h

```
#include <iostream>

#include <math.h>

#define _USE_MATH_DEFINES

using namespace std;

float s_calculation(float x, float y, float z)
{
    float s;

    s = log(x - y)+sqrt((M_PI * pow(x, 2))/(x + (z/2 * pow(y,2))));

    return roundf(s * 100000.0f) / 100000.0f;
}

string task9_1(int point)
{
    if (point >= 0 && point <=12)
    {
        switch (point)
        {
            case 0:
                return "<1.08";
            case 1:
                return "1.08-5.4";
            case 2:
                return "5.76-33.84";
            case 3:
                return "33.84-19.44";
            case 4:
                return "19.8-28.4";
            case 5:
                return "28.8-38.52";
```

```

        case 6:
            return "38.88-49.68 ";
        case 7:
            return "50.04-61.56";
        case 8:
            return "61.92-74.52";
        case 9:
            return "74.88-87.84";
        case 10:
            return "88.2-102.24";
        case 11:
            return "102.6-117.36";
        case 12:
            return ">=117.72";
    }
}

```

```
float task9_2(float footsize)
```

```

{
    if(footsize == 35)
        return 23;
    if(footsize == 36)
        return 23.5;
    if(footsize == 36.5)
        return 24;
    if (footsize == 37)
        return 24.5;
    if (footsize == 38)
        return 25;
    if (footsize == 39)
        return 25.5;
    if (footsize == 40.5)
        return 26;
    if (footsize == 41)
        return 26.5;
    if (footsize == 41.5)

```

```

        return 27;
    if (footsize == 42)
        return 27.5;
    if (footsize == 43)
        return 28.5;
    if (footsize == 44)
        return 29;
    if (footsize == 45)
        return 29.5;
    if (footsize == 46)
        return 30;
    if (footsize == 47)
        return 30.5;
    if (footsize == 47.5)
        return 31;
    if (footsize == 48)
        return 31.5;
    if (footsize == 48.5)
        return 32;
}

```

```

int task9_3(int number)
{
    if (number >=1000 && number <=10000)
    {
        int bit = (number >> 14) & 1;
        int count = 0;

        if (bit == 0)
        {
            while (number != 0)
            {
                count += (number & 1) == 0 ? 1 : 0;
                number >>= 1;
            }
            return count;
        }
    }
}

```

```

else
{
    while (number != 0)
    {
        count += (number & 1) == 1 ? 1 : 0;
        number >>= 1;
    }
    return count;
}
}

```

Додаток 5 -Лістинг вихідного коду TestDriver

```

#include <iostream>

#include "ModulesChervonyi.h"

using namespace std;

int main()
{
    cout << "Unit testing for task 9.1\n";
    for(int i = 0; i < 4; i++)
    {
        int p;
        string result;
        cout << "Enter Bophor point: ";
        cin >> p;
        cout << "Result: ";
        cin >> result;
        if (task9_1(p) == result)
            cout << "Test#"<<i+1<<" Passed\n";
        else
            cout << "Test#"<<i+1<<" Failed\n";
    }

    cout << "Unit testing for task 9.2\n";
    for (int i = 0; i < 4; i++)
    {
        float footsize;

```

```

        float result;

        cout << "Enter Ukraine foot size: ";
        cin >> footsize;
        cout << "Result: ";
        cin >> result;
        if (task9_2(footsize) == result)
            cout << "Test#" << i + 1 << " Passed\n";
        else
            cout << "Test#" << i + 1 << " Failed\n";
    }

    cout << "Unit testing for task 9.3\n";
    for (int i = 0; i < 4; i++)
    {
        int number;
        int result;
        cout << "Enter number: ";
        cin >> number;
        cout << "Result: ";
        cin >> result;
        if (task9_3(number) == result)
            cout << "Test#" << i + 1 << " Passed\n";
        else
            cout << "Test#" << i + 1 << " Failed\n";
    }
}

```

Висновок

Завантажений власний Git-репозиторій <https://github.com/odorenskyi/Chervonyi-Yehor-KN22>. У \Lab9 заповнено файл README.md, створено теки prj, Software, TestSuite, Report. До звіту включено тему мету завдання. Здійснено аналіз і постановку задач 9.1, 9.2, 9.3. Виконано аналіз вимог, проектування архітектури, детальне проектування програмних модулів розв'язування задач 9.1, 9.2, 9.3. Розроблено набори контрольних прикладів до задач 9.1, 9.2, 9.3 задля виконання модульного тестування (Unit testing) модулів C++. В Code::Blocks IDE відкрито проект статичної бібліотеки ModulesChervonyi з \Lab8\prj, створений під час виконання лабораторної роботи № 8. За отриманими під час проектування програмних модулів артефактами виконано конструювання функцій: мовою програмування C++ реалізовано функції, які за наданим інтерфейсом реалізують розв'язування задач 9.1, 9.2 та 9.3 відповідно. Скомпільовано ModulesChervonyi. Відкрито проект заголовкового файлу ModulesChervonyi із \Lab8\prj, створений під час виконання лабораторної роботи № 8, та доповнено його прототипами функцій розв'язування задач 9.1–9.3. В Code::Blocks IDE у \prj створено проект консольного додатка C++, під назвою TestDriver. Реалізовано тестовий драйвер для виконання розроблених тестових наборів до задач 9.1-9.3 і за його допомогою виконано модульне тестування функцій зі статичної бібліотеки libModulesChervonyi.a. Результати unit-тестування задокументовано. Вихідний код проектів ModulesChervonyi та TestDriver включено до звіту як додатки. Здійснено аналіз і постановку задачі 9.4. Виконано аналіз вимог, проектування архітектури, детальне проектування програмного забезпечення розв'язування задачі 9.4. Розроблено тест-сют для виконання системного тестування ПЗ розв'язування задачі 9.4. В Code::Blocks IDE створено проект консольного додатка Chervonyi_task у теці \prj. Мовою програмування C++ реалізовано результати проектування програмного забезпечення розв'язування задачі 9.4. Консольний додаток Chervonyi_task.exe скопійовано у \Software. В ході роботи над лабораторною роботою набув 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.