

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.

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

ВИКОНАВ
студент академічної групи ТК-23-1
Шавлєнков П.О

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

Тема: Реалізація програмних модулів, розгалужених та ітераційних обчислювальних процесів.

Мета: Набуття ґрунтовних вмінь і практичних навичок реалізації технології модульного програмування, застосування операторів 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:

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

ВАРІАНТ 1

— ЗАДАЧА 9.1 —

Вхід: швидкість вітру (км/год) під час торнадо.

Вихід: категорія торнадо за шкалою Фудзіті та частота їх виникнення.



КАТЕГОРІЯ	ШВИДКІСТЬ ВІТРУ, км/год	ЧАСТОТА
F0	64 – 116	38,9 %
F1	117 – 180	35,6 %
F2	181 – 253	19,4 %
F3	254 – 332	4,9 %
F4	333 – 418	1,1 %
F5	419 – 512	менше 0,1 %

— ЗАДАЧА 9.2 —

Вхід: температура повітря (у градусах за шкалою Цельсія), зафіксовану о 00:00, 04:00, 8:00, 12:00, 16:00, 20:00 год.

Вихід: середньодобова температура за шкалами Цельсія та Фаренгейта.



Знаючи температуру за шкалою Цельсія, температуру за шкалою Фаренгейта можна розрахувати наступним чином:

$$t_F = 32 + \frac{9}{5}t_C,$$

де t_F – температура за шкалою Фаренгейта, t_C – температура за шкалою Цельсія.

— ЗАДАЧА 9.3 —

Вхід: натуральне число N від 0 до 65535.

Вихід: якщо біт D_0 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 “z” викликається `s_calculation()`, “t” – функція задачі 9.1, “s” – функція задачі 9.2, “i” – 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 “q” або “Q”, відбувається вихід з програми, інакше — виконання програми повторюється.



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

Варіант 1

Завдання 9.1

Вхід: швидкість вітру (км/год) під час торнадо.

Вихід: категорія торнадо за шкалою Фудзіти та частота їх виникнення.



КАТЕГОРІЯ	ШВИДКІСТЬ ВІТРУ, км/год	ЧАСТОТА
F0	64 – 116	38,9 %
F1	117 – 180	35,6 %
F2	181 – 253	19,4 %
F3	254 – 332	4,9 %
F4	333 – 418	1,1 %
F5	419 – 512	менше 0,1 %

Вхідні дані: швидкість вітру (км/год) під час торнадо.

Вихідні дані: категорія торнадо за шкалою Фудзіти та частота їх виникнення.

Алгоритм:

1. Підключаємо статичну бібліотеку модулів ModulesShavlienkov.h
2. Запитуємо у користувача speed.
3. Викликаємо функцію printInfoTornado в яку передаємо x, y, z та виводимо результат.

Лістинг:

ModulesShavlienkov/ModulesShavlienkov.cpp

```
void printInfoTornado(float speed) {  
    if(64 <= speed && 116 >= speed) {  
        cout << "Категорія торнадо за шкалою Фудзіти: " << "F0" << endl << "Частота  
виникнення: 38,9%" << endl;  
    } else if(117 <= speed && 180 >= speed) {  
        cout << "Категорія торнадо за шкалою Фудзіти: " << "F1" << endl << "Частота  
виникнення: 35,6%" << endl;  
    } else if(181 <= speed && 253 >= speed) {  
        cout << "Категорія торнадо за шкалою Фудзіти: " << "F2" << endl << "Частота  
виникнення: 19,4%" << endl;  
    } else if(254 <= speed && 332 >= speed) {  
        cout << "Категорія торнадо за шкалою Фудзіти: " << "F3" << endl << "Частота  
виникнення: 4,9%" << endl;  
    } else if(333 <= speed && 418 >= speed) {  
        cout << "Категорія торнадо за шкалою Фудзіти: " << "F4" << endl << "Частота  
виникнення: 1,1%" << endl;  
    } else if(419 <= speed && 512 >= speed) {  
        cout << "Категорія торнадо за шкалою Фудзіти: " << "F5" << endl << "Частота  
виникнення: менше 0,1%" << endl;  
    } else {  
        cout << "Не входить у діапазон" << endl;  
    }  
}
```

```
}  
}
```

ModulesShavlienkov.h

```
#ifndef MODULESSHAVLIENKOV_H_INCLUDED  
#define MODULESSHAVLIENKOV_H_INCLUDED  
  
float s_calculation(int x, int y, int z);  
void printInfoTornado(float speed);  
void avgTemp(float arr[]);  
void countBits(int N);  
  
#endif // MODULESSHAVLIENKOV_H_INCLUDED
```

TestDriver/main.cpp

```
float speed;  
  
cout << "Введіть швидкість вітру: ";  
cin >> speed;  
  
printInfoTornado(speed);
```

Завдання 9.2

Вхід: температура повітря (у градусах за шкалою Цельсія), зафіксовану о 00:00, 04:00, 8:00, 12:00, 16:00, 20:00 год.

Вихід: середньодобова температура за шкалами Цельсія та Фаренгейта.



Знаючи температуру за шкалою Цельсія, температуру за шкалою Фаренгейта можна розрахувати наступним чином:

$$t_F = 32 + \frac{9}{5}t_C,$$

де t_F – температура за шкалою Фаренгейта, t_C – температура за шкалою Цельсія.

Вхідні дані: температура повітря (у градусах за шкалою Цельсія), зафіксовану о 00:00, 04:00, 08:00, 12:00, 16:00, 20:00 год.

Вихідні дані: середньодобова температура за шкалами Цельсія та Фаренгейта.

Алгоритм:

1. Підключаємо статичну бібліотеку модулів ModulesShavlienkov.h
2. Запитуємо у користувача масив temps[6]
3. Викликаємо функцію avgTemp в яку передаємо temps[6] та виводимо результат.

Лістинг:

ModulesShavlienkov/ModulesShavlienkov.cpp

```
void avgTemp(float arr[]) {  
  
    float sum_temp = 0;  
  
    for(int i = 0; i < 6; i++) {  
        sum_temp += arr[i];  
    }  
  
    cout << "Цельсій: " << (sum_temp / 6) << " °C" << endl;  
    cout << "Фаренгейт: " << ((sum_temp / 6) * (9 / 5)) + 32 << " °F" << endl;  
}
```

ModulesShavlienkov.h

```
#ifndef MODULESSHAVLIENKOV_H_INCLUDED  
#define MODULESSHAVLIENKOV_H_INCLUDED  
  
float s_calculation(int x, int y, int z);  
void printInfoTornado(float speed);  
void avgTemp(float arr[]);  
void countBits(int N);  
  
#endif // MODULESSHAVLIENKOV_H_INCLUDED
```

TestDriver/main.cpp

```
float temps[6];  
  
for(int i = 0; i < 6; i++) {  
    float temp;  
    cout << "Введіть температуру: ";  
    cin >> temp;  
    temps[i] = temp;  
}  
  
avgTemp(temps);
```

Завдання 9.3

Вхід: натуральне число N від 0 до 65535.

Вихід: якщо біт D_0 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 « ? : ».

Вхідні дані: натуральне число N від 0 до 65535.

Вихідні дані: якщо біт D_0 числа N рівний 0, кількість двійкових нулів у ньому, інакше — кількість двійкових одиниць.

Алгоритм:

1. Підключаємо статичну бібліотеку модулів ModulesShavlienkov.h
2. Запитуємо у користувача число N .
3. Викликаємо функцію countBits в яку передаємо N та виводимо результат.

Лістинг:

ModulesShavlienkov/ModulesShavlienkov.cpp

```
void countBits(int N) {  
  
    int counter = 0;  
  
    if((N & 1) == 0) {  
  
        while (N > 0) {  
            int bit = N % 2;  
            counter += (bit == 0) ? 1 : 0;  
            N /= 2;  
        }  
  
        cout << "Кількість двійкових нулів дорівнює: " << counter << endl;  
  
    } else {  
  
        while (N > 0) {  
            int bit = N % 2;  
            counter += (bit != 0) ? 1 : 0;  
            N /= 2;  
        }  
  
        cout << "Кількість двійкових одиниць дорівнює: " << counter << endl;  
    }  
}
```

ModulesShavlienkov.h

```
#ifndef MODULESSHAVLIENKOV_H_INCLUDED
#define MODULESSHAVLIENKOV_H_INCLUDED

float s_calculation(int x, int y, int z);
void printInfoTornado(float speed);
void avgTemp(float arr[]);
void countBits(int N);

#endif // MODULESSHAVLIENKOV_H_INCLUDED
```

TestDriver/main.cpp

```
int N;

cout << "Введіть число: ";
cin >> N;

countBits(N);
```

Завдання 9.4

За введенням користувачем символом “z” викликається `s_calculation()`, “r” – функція задачі 9.1, “s” – функція задачі 9.2, “t” – 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 “q” або “Q”, відбувається вихід з програми, інакше — виконання програми повторюється.

Вхідні дані: Символ z, r, s, t, q, Q.

Вихідні дані: Якщо користувач вводить символ “z” викликається `s_calculation()`, “r” – функція задачі 9.1, “s” – функція задачі 9.2, “t” – функція задачі 9.3. Якщо користувач вводить інший символ – видати звуковий сигнал про помилкове введення та продовжити цикл. Якщо користувач вводить “q” або “Q” – зробити вихід з програми.

Алгоритм:

1. Підключаємо статичну бібліотеку модулів `ModulesShavlienkov.h`
2. Викликаємо нескінченний цикл `while`.
3. Запитуємо у користувача символ (z, r, s, t, q, Q).
4. Якщо користувач вводить символ “z” викликаємо функцію `s_calculation()`, “r” – функція задачі 9.1, “s” – функція задачі 9.2, “t” – функція задачі 9.3. Якщо користувач вводить інший символ – видати звуковий сигнал про помилкове введення та продовжити цикл. Якщо користувач вводить “q” або “Q” – зробити вихід з програми.


```

        cin >> temp;
        temps[i] = temp;
    }

    avgTemp(temps);

    break;

case 't':
    int N;

    cout << "Введіть число: ";
    cin >> N;

    countBits(N);

    break;

default:
    Beep(523, 1000);

    break;
}

}

return 0;
}

```

Висновок:

У ході лабораторної роботи я успішно вивчив основні принципи модульного програмування та вмю їх застосовувати у практиці. Також я отримав поглиблені знання з операторів мови програмування C++, зокрема арифметичних, логічних та побітових операцій. Засвоївши концепції умов, циклів та вибору, я можу ефективно використовувати їх у своїх програмах для забезпечення правильної роботи та оптимізації коду.