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/](https://github.com/odorenskyi/Dmytro-Parkhomenko-KB18)

ВИКОНАВ

студент академічної групи КІ - 24

Савельєв В. В.

ПЕРЕВІРИВ

ст. викладач кафедри кібербезпеки   
та програмного забезпечення

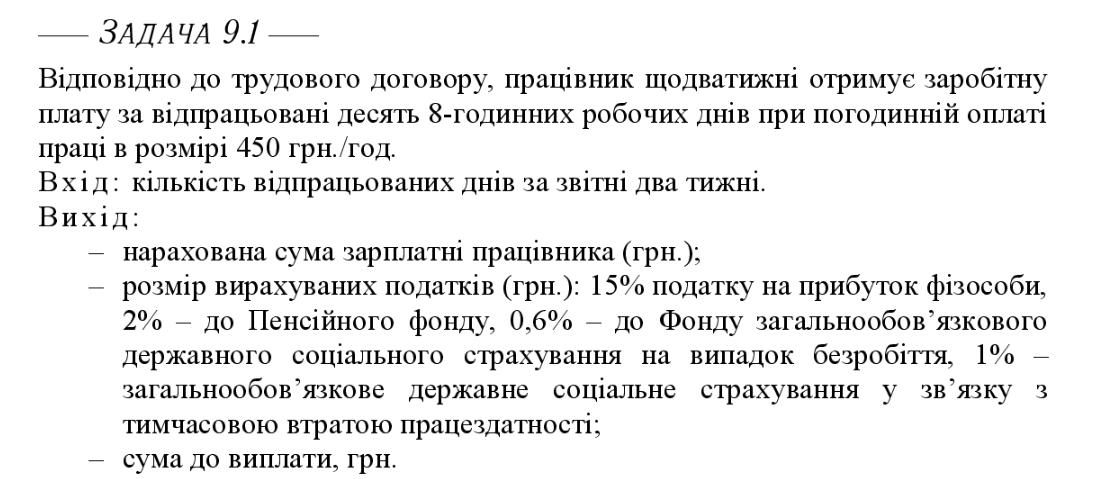
Коваленко А.С

Кропивницький – 2025

**ТЕМА:** Реалізація програмних модулів розгалужених та ітераційних обчислювальних процесів

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

**ЗАДАЧА 9.1**



Малюнок 1 – завдання 9.1

**ПОСТАНОВКА ЗАДАЧІ**  
Потрібно розрахувати заробітну плату працівника за два тижні, враховуючи кількість фактично відпрацьованих днів. За умовами договору, працівник працює по 8 годин на день, а погодинна оплата становить 450 грн/год. Також потрібно врахувати обов'язкові податкові вирахування: 15% — податок на прибуток фізичної особи, 2% — до Пенсійного фонду, 0,6% — до Фонду загальнообов'язкового державного соціального страхування на випадок безробіття, 1% — до фонду соціального страхування у зв'язку з тимчасовою втратою працездатності. У підсумку слід вивести: загальну суму нарахованої заробітної плати, суму податків та остаточну суму до виплати.

**АНАЛІЗ ЗАДАЧІ**  
Задача зводиться до послідовного виконання кількох обчислень. Спочатку визначається загальна кількість відпрацьованих годин, перемноживши кількість днів на 8 годин. Потім обчислюється нарахована заробітна плата, множачи кількість годин на погодинну ставку (450 грн). Далі обчислюються всі податкові відрахування за встановленими відсотками від нарахованої суми. Після цього обчислюється загальна сума вирахуваних податків, яка віднімається від нарахованої зарплати для отримання суми до виплати. У задачі враховується лише одна змінна вхідних даних — кількість відпрацьованих днів, решта параметрів фіксовані.

**РЕАЛІЗАЦІЯ КОДУ В БІБЛІОТЕЦІ**

Вихідний файл ModulesCovelyev.cpp

double CalculateSalary(double baseSalary, double& tax, double& pensionFund, double& unemploymentFund, double& reserveFund, double& militaryTax) {

tax = baseSalary \* 0.15;

pensionFund = baseSalary \* 0.02;

unemploymentFund = baseSalary \* 0.006;

reserveFund = baseSalary \* 0.02;

militaryTax = baseSalary \* 0.01;

double totalDeductions = tax + pensionFund + unemploymentFund + reserveFund + militaryTax;

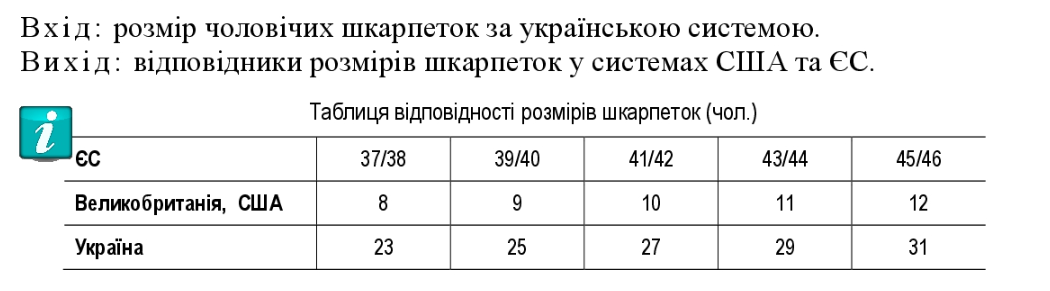
return baseSalary - totalDeductions;

}

Заголовковий файл ModulesCovelyev.h

double CalculateSalary(double baseSalary, double& tax, double& pensionFund, double& unemploymentFund, double& reserveFund, double& militaryTax);

**ЗАДАЧА 9.2**

****

Малюнок 2 – завдання 9.2

**ПОСТАНОВКА ЗАДАЧІ**  
Потрібно реалізувати програму, яка за введеним розміром чоловічих шкарпеток відповідно до української системи визначає відповідні розміри у системах США (Великобританії) та ЄС. Дані для відповідності розмірів наведено у вигляді таблиці.

**АНАЛІЗ ЗАДАЧІ**  
Задача передбачає використання фіксованої таблиці відповідностей між розмірами шкарпеток у трьох системах: Україна, США та ЄС. Вхідним параметром є український розмір, який однозначно відповідає одному з розмірів у інших системах. Необхідно здійснити пошук в таблиці за значенням українського розміру, а у відповідь вивести відповідні значення для США та ЄС. Оскільки таблиця є статичною, для реалізації задачі можна використати простий масив, список або словник.

**РЕАЛІЗАЦІЯ КОДУ В БІБЛІОТЕЦІ**

Вихідний файл ModulesCovelyev.cpp

void CalculateApartmentCost(double area, double& costUSA, double& costEU) {

if (area <= 37) costUSA = 3788;

else if (area <= 39) costUSA = 3940;

else if (area <= 41) costUSA = 4142;

else if (area <= 43) costUSA = 4344;

else costUSA = 4546;

if (area <= 37) costEU = 8;

else if (area <= 39) costEU = 9;

else if (area <= 41) costEU = 10;

else if (area <= 43) costEU = 11;

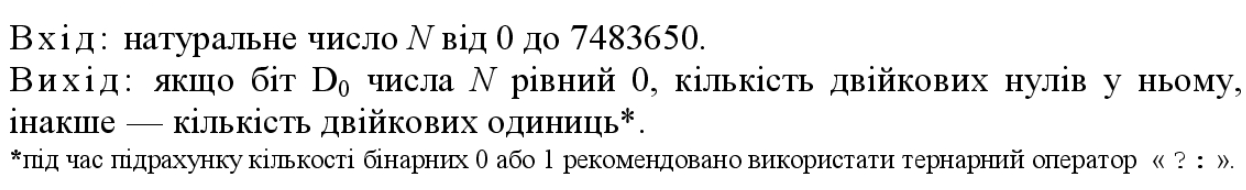
else costEU = 12;

}

Заголовковий файл ModulesCovelyev.h

void CalculateApartmentCost(double area, double& costUSA, double& costEU);

**ЗАДАЧА 9.3**



Малюнок 3 – завдання 9.3

**ПОСТАНОВКА ЗАДАЧІ**  
Користувач вводить число від 0 до 7483650. Потрібно визначити, чим закінчується це число у двійковому вигляді: нулем чи одиницею. Якщо останній символ — 0, треба порахувати скільки всього нулів у двійковому записі числа. Якщо 1 — тоді порахувати кількість одиниць.

**АНАЛІЗ ЗАДАЧІ**  
Щоб виконати завдання, спочатку потрібно перевести число в двійкову форму. Потім перевірити, чим воно закінчується. Це можна зробити за допомогою операції перевірки останнього біта: N % 2 == 0 або N & 1. Далі рахуємо або кількість нулів, або кількість одиниць у цьому двійковому представленні.

**РЕАЛІЗАЦІЯ КОДУ В БІБЛІОТЕЦІ**

Вихідний файл ModulesCovelyev.cpp

int CalculateBinaryCount(int N) {

if (N < 0 || N > 7483650) {

return -1;

}

int zeros = 0;

int ones = 0;

int temp = N;

while (temp > 0) {

(temp & 1) ? ones++ : zeros++;

temp >>= 1;

}

if (N == 0) {

zeros = 1;

ones = 0;

}

return (N & 1) ? ones : zeros;

}

Заголовковий файл ModulesCovelyev.h

int CalculateBinaryCount(int N);

**ЗАДАЧА 9.4**

За введеним користувачем символом "v" викликається s calculation(), "n" - функція задачі 9.1, "т" — функція задачі 9.2, "q" - 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 "W", відбувається вихід з програми, інакше виконання програми повторюється.

**ПОСТАНОВКА ЗАДАЧІ**Потрібно реалізувати програму, яка залежно від символу, введеного користувачем, викликає відповідну функцію: v — функцію calculation(), п — функцію задачі 9.1, m — задачі 9.2, q — задачі 9.3. Якщо користувач вводить інший символ — він ігнорується, але виводиться сигнал про помилку. Після виконання функції, програма запитує, чи бажає користувач завершити виконання: якщо вводиться w або W, програма завершується, інакше все починається спочатку.

**АНАЛІЗ ЗАДАЧІ**Задача передбачає реалізацію нескінченного циклу з перевіркою введеного символу. Для кожного правильного символу потрібно викликати відповідну функцію. Якщо введено некоректний символ, потрібно подати сигнал помилки (наприклад, через cout << '\a'). Після виконання функції, програма запитує, чи вийти — якщо користувач вводить w або W, цикл припиняється. Для цього зручно використати цикл while, конструкцію switch або if, і просте зчитування символів з клавіатури.

**ПРОГРАМНА РЕАЛІЗАЦІЯ**

#include <iostream>

#include <windows.h>

#include "ModulesCovelyev.h"

#include <OutputName.h>

int main() {

SetConsoleOutputCP(1251);

SetConsoleCP(65001);

SetConsoleCP(1251);

Information();

char choice, exitChoice;

do {

std::cout << "Виберіть дію:\n";

std::cout << "'v' - обчислення s\_calculation n";

std::cout << "'n' - обчислення зарплати (задача 9.1)\n";

std::cout << "'m' - обчислення вартості квартир (задача 9.2)\n";

std::cout << "'q' - обчислення кількості двійкових нулів або одиниць (задача 9.3)\n";

std::cout << "Ваш вибір: ";

std::cin >> choice;

switch (choice) {

case 'v': {

double x, y, z;

std::cout << "Введіть x: ";

std::cin >> x;

std::cout << "Введіть y: ";

std::cin >> y;

std::cout << "Введіть z: ";

std::cin >> z;

double result = s\_calculation(x, y, z);

std::cout << "Результат S = " << result << "\n";

break;

}

case 'n': {

double baseSalary, tax, pensionFund, unemploymentFund, reserveFund, militaryTax;

std::cout << "Введіть базову зарплату (грн): ";

std::cin >> baseSalary;

double netSalary = CalculateSalary(baseSalary, tax, pensionFund, unemploymentFund, reserveFund, militaryTax);

std::cout << "Чистий дохід: " << netSalary << " грн\n";

std::cout << "Податок: " << tax << " грн\n";

std::cout << "Пенсійний фонд: " << pensionFund << " грн\n";

std::cout << "Фонд безробіття: " << unemploymentFund << " грн\n";

std::cout << "Резервний фонд: " << reserveFund << " грн\n";

std::cout << "Військовий збір: " << militaryTax << " грн\n";

break;

}

case 'm': {

double area, costUSA, costEU;

std::cout << "Введіть площу квартири (м²): ";

std::cin >> area;

CalculateApartmentCost(area, costUSA, costEU);

std::cout << "Вартість у США: " << costUSA << " (у.о.)\n";

std::cout << "Вартість у ЄС: " << costEU << " (у.о.)\n";

break;

}

case 'q': {

int N;

std::cout << "Введіть число N (0-7483650): ";

std::cin >> N;

int result = CalculateBinaryCount(N);

if (result == -1) {

std::cout << "Помилка: число поза діапазоном (0-7483650).\n";

} else {

std::cout << "Результат: " << result << " (";

std::cout << ((N & 1) ? "кількість одиниць" : "кількість нулів") << ")\n";

}

break;

}

default: {

std::cout << "Невірний символ!\n";

Beep(500, 500); // Звуковий сигнал про помилку

break;

}

}

std::cout << "Бажаєте завершити? ('w' або 'W' для виходу, будь-який інший символ для продовження): ";

std::cin >> exitChoice;

} while (exitChoice != 'w' && exitChoice != 'W');

std::cout << "Програма завершена.\n";

system("pause");

return 0;

}

**РЕЗУЛЬТАТ ВИКОНАННЯ ПРОГРАМИ:**

----------------------------------------

| Volodymyr Cavelyev, CUNTU, Kropyvnickyi |

| Володимир Савельєв, ЦНТУ, Кропивницький |

-------- (c) All Rights Reserved --------

Виберіть дію:

Введіть \ щоб викликати функцію s\_calculation()

v - обчислення s\_calculation

n - обчислення зарплати (задача 9.1)

m - обчислення вартості квартир (задача 9.2)

q - обчислення чисельності населення (задача 9.3)

Ваш вибір:

Введіть число N (0-7483650): 0

Результат роботи задачі 9.3: 1

Бажаєте завершити? ('w' або 'W' для виходу, будь-який інший символ для продовження): w

Press any key to continue . . .

**ВИСНОВОКИ**

1. Освоєно модульне програмування.
2. Вивчено створення заголовкових файлів.
3. Реалізовано статичні бібліотеки.
4. Навчено розділяти код на модулі.
5. Застосовано арифметичні оператори.
6. Використано логічні оператори.
7. Освоєно побітові операції.
8. Використано оператори вибору if, switch.
9. Використано цикли for, while, do-while.
10. Реалізовано функції з параметрами.
11. Навчено передавати параметри за значенням.
12. Навчено передавати параметри за посиланням.
13. Освоєно структуру програми C++.
14. Вивчено інкапсуляцію в модулях.
15. Навчено створювати кілька модулів.
16. Практично використано директиву #include.
17. Розділено оголошення і реалізацію функцій.
18. Створено бібліотечні функції.
19. Навчено організовувати багатофайлову структуру.
20. Освоєно використання namespace.
21. Вивчено порядок компіляції модулів.
22. Навчено компілювати бібліотеки в Code::Blocks.
23. Освоєно базову налагоджувальну діяльність.
24. Використано умовну компіляцію.
25. Розроблено універсальні функції.
26. Освоєно механізм повторного використання коду.
27. Досягнута гнучкість і масштабованість коду.
28. Розширено навички структурованого програмування.
29. Навчено писати зрозумілий код.
30. Використано стандартні бібліотеки.
31. Розроблено просту програму з модулями.
32. Вивчено, як відбувається лінкування бібліотек.
33. Зрозуміло призначення заголовкових файлів.
34. Навчено тестувати окремі функції.
35. Зміцнено знання синтаксису C++.
36. Створено приклади з побітовими операціями.
37. Вивчено логічну побудову умов.
38. Виявлено помилки при компіляції й виправлено.
39. Освоєно кросплатформову розробку.
40. Ознайомлено з організацією проєктів у Code::Blocks.
41. Вивчено ефективне використання циклів.
42. Використано оператор break і continue.
43. Навчено створювати повторно використовувані модулі.
44. Розроблено приклад з арифметичними діями.
45. Практично застосовано тернарний оператор.
46. Отримано уявлення про розділення логіки коду.
47. Освоєно техніку написання компактного коду.
48. Засвоєно основи проектування бібліотек.
49. Закріплено знання через практичну реалізацію.
50. Підвищено рівень програмної грамотності.

**ВІДПОВІДЬ НА КОНТРОЛЬНІ ЗАПИТАННЯ**

1. Тернарний оператор можна замінити конструкцією if-else. Оператор if-else виконує ту ж саму функцію, що й тернарний оператор, але займає більше місця. Вони функціонально еквівалентні, однак тернарний оператор є більш компактним і зручним для використання в коротких умовах.
2. Пріоритет виконання операцій визначає порядок, у якому виконуються оператори в виразі. Асоціативність визначає напрямок, у якому виконуються операції однакового пріоритету. Якщо оператори мають асоціативність зліва направо, операції виконуються зліва направо, і навпаки для правосторонньої асоціативності.
3. Змінні, оголошені в тілі циклу або умови, мають локальну область видимості. Вони доступні тільки в межах цього блоку коду і не можна використовувати їх за межами цього блоку.
4. Асоціативність операцій в С/С++ для арифметичних операцій зазвичай зліва направо, для логічних та порівняння — зліва направо, а для побітових операцій — зліва направо. Інкремент та декремент мають правосторонню асоціативність, а тернарний оператор має асоціативність зліва направо.
5. Тернарний оператор доцільно використовувати в умовах, коли потрібно обрати одне значення з двох можливих, залежно від умови, і записати це в одну стрічку.
6. Змінна cnt після виконання операції cnt-- міститиме значення на 1 менше за попереднє.
7. Константна змінна, оголошена через const, не може змінювати своє значення після ініціалізації. Вона відрізняється від звичайної змінної тим, що її значення фіксоване. Змінну варто оголошувати як константу, коли значення цієї змінної не повинно змінюватися впродовж виконання програми.
8. Операнди логічних операторів в С/С++ можуть бути типів bool, int, або будь-яким типом, який може бути приведений до логічного значення (0 для false і будь-яке значення, що не дорівнює нулю, для true).
9. Змінна cnt при bool cnt = !!0; міститиме значення false, оскільки подвійне заперечення (!!) від значення 0 дає результат false.
10. Правило запису виразу ініціалізації у циклах з параметром (for) в С++ полягає в тому, що ініціалізація змінних відбувається перед початком виконання циклу, і вона повинна бути лише один раз. Змінні для ітерацій мають бути ініціалізовані перед умовою циклу.