

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет
«Харківський авіаційний інститут»

Факультет систем управління літальних апаратів
Кафедра систем управління літальних апаратів

Лабораторна робота № 2

з дисципліни «Алгоритмізація та програмування»
на тему «"Математичні обчислення на мові C ++"»

XAI.301.G5.310.4 ЛР

Виконав студент гр. _____310_____

25.09.2025

(підпис, дата)

Дмитро ДОБРОВЕЦЬКИЙ

(П.І.Б.)

Перевірів

_____ к.т.н., доц. Олена ГАВРИЛЕНКО

(підпис, дата)

(П.І.Б.)

2025

МЕТА РОБОТИ

Вивчити теоретично базові типи даних мови C ++ і реалізувати консольний додаток лінійної структури для введення / виведення і обробки змінних базових типів з використанням вбудованих операцій та бібліотечних функцій на мові програмування C ++.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вирішити задачу з цілочисельними змінними. Всі вхідні і вихідні дані в задачах цієї групи є цілими числами. Всі числа, для яких вказано кількість цифр (двозначне число, тризначне число і т. д.), вважаються додатними. Integer 37.

Завдання 2. Вирішити завдання з логічними змінними. У всіх завданнях даної групи потрібно вивести логічне значення true (1), якщо наведене висловлювання для запропонованих вхідних даних є істинним, і значення false (0) в іншому випадку. Всі числа, для яких вказано кількість цифр (двозначне число, тризначне число і т. д.), вважаються цілими додатними. Boolean 16.

Завдання 3. Обчислити математичний вираз зі змінними дійсного типу, використовуючи стандартну бібліотеку cmath. Число π має бути визначено як константа дійсного типу. Вираз 25.

Завдання 4. Використовуючи ChatGpt, Gemini або інший засіб генеративного ШІ, провести самоаналіз отриманих знань і навичок за допомогою наступних промптів:

«Ти - викладач, що приймає захист моєї роботи. Задай мені 5 тестових питань з 4 варіантами відповіді і 5 відкритих питань - за кодом, що є у файлі звіту і теоретичними відомостями у файлі лекції»

«Оціни повноту, правильність, час відповіді та ймовірність використання штучного інтелекту для кожної відповіді. Сформулюй загальну оцінку у 10-бальній шкалі, віднімаючи 50% балів там, де ймовірність відповіді з засобом ШІ висока»

Додайте 2-3 власних промпта для прояснення питань чи коментарів до оцінок ваших відповідей

ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі Integer 37

Лістинг коду вирішення задачі Begin37 наведено в дод. А (стор. 4).

Екрани робіт програми показаний на рис. Б.1.

Завдання 2.

Вирішення задачі Boolean 16

Лістинг коду вирішення задачі Begin16 наведено в дод. А (стор. 4).

Екрани робіт програми показаний на рис. Б.1.

Завдання 2.

Вирішення задачі Вираз 25

Лістинг коду вирішення задачі Begin25 наведено в дод. А (стор. 4).

Екрани робіт програми показаний на рис. Б.1.

Завдання 2.

2. Діалог з Chat GPT для самоаналізу наведено у дод. В.

ВИСНОВКИ

Було вивчено базові типи даних мови C++ та їх практичне застосування у консольних додатках. На практиці закріплено навички роботи з цілочисельними, дійсними та логічними змінними, включаючи використання стандартних операцій і бібліотечних функцій. Отримано досвід написання програм лінійної структури для обробки даних різних типів та організації взаємодії з користувачем.

ДОДАТОК А

Лістинг коду програми

```
#include <iostream>
#include <cmath> // підключення бібліотеки математичних функцій
using namespace std;

int main()
{
    // Integer37. Дано прямокутник довжиною А і шириною В.
    // Знайти, скільки квадратних плиток зі стороною С можна вмістити
    // та яка частина прямокутника залишиться незаповненою?

    cout << "Integer37." << endl;

    int A, B, C; // довжина, ширина прямокутника та сторона плитки
    int tiles_x, tiles_y, total_tiles; // кількість плиток по ширині, довжині та
загальна
    int area_rectangle, area_tiles, area_remaining; // площі

    // введення даних
    cout << "А (довжина прямокутника) = ";
    cin >> A;
    cout << "В (ширина прямокутника) = ";
    cin >> B;
    cout << "С (сторона плитки) = ";
    cin >> C;

    // перевірка на коректність введених даних
    if (A <= 0 || B <= 0 || C <= 0)
    {
        cout << "Помилка: всі значення повинні бути додатними!" << endl;
        return 1;
    }

    // обчислення кількості плиток
    tiles_x = A / C; // кількість плиток по довжині
    tiles_y = B / C; // кількість плиток по ширині
    total_tiles = tiles_x * tiles_y; // загальна кількість плиток

    // обчислення площ
    area_rectangle = A * B; // площа прямокутника
    area_tiles = total_tiles * C * C; // площа, зайнята плитками
    area_remaining = area_rectangle - area_tiles; // незаповнена площа

    // виведення результатів
    cout << "Кількість плиток: " << total_tiles << endl;
    cout << "Незаповнена площа: " << area_remaining << endl << endl;
```

```

// Boolean16. Дано ціле додатне число.
// Перевірити істинність висловлювання:
// «Дане число є парним двозначним».

cout << "Boolean16." << endl;

int number;
bool result;

// введення даних
cout << "Введіть ціле додатне число: ";
cin >> number;

// перевірка істинності висловлювання
// Число є двозначним: >= 10 та <= 99
// Число є парним: ділиться на 2 без остачі
result = (number >= 10) && (number <= 99) && (number % 2 == 0);

// виведення результату
cout << "Число " << number << " є парним двозначним: " << boolalpha << result
<< endl << endl;

// y = ... (tab.3 N25)
// Обчислення математичного виразу
//  $y = (\sqrt[5]{x^6} + \sqrt{2|x|}) / \log_3(\sin^2(|2x| + 29^\circ)) + (\pi|3\operatorname{tg}(x^2)|)/5$ 

cout << "Обчислення математичного виразу" << endl;

const double pi = 3.141592653589793; // визначення константи п
double x, y; // декларація дійсних змінних

// введення даних
cout << "Дійсний аргумент x = ";
cin >> x;

// обчислення чисельника першого доданку
double term1_num = pow(fabs(x), 6.0/5.0) + sqrt(2 * fabs(x));

// обчислення знаменника першого доданку
double angle_rad = (fabs(2 * x) + 29.0) * pi / 180.0; // переведення градусів
у радіани
double sin_value = sin(angle_rad);
double log_argument = pow(sin_value, 2);
double term1_denom = log(log_argument) / log(3.0); //  $\log_3(a) = \ln(a)/\ln(3)$ 

// перший доданок

```

```
double term1 = term1_num / term1_denom;

// другий доданок
double term2 = (pi * fabs(3 * tan(pow(x, 2)))) / 5.0;

// підсумковий результат
y = term1 + term2;

// виведення результату
cout << "Функція y = " << y << endl << endl;

return 0;
}
```

ДОДАТОК Б
Скрін-шоти вікна виконання програми

```
Integer37.  
A (довжина прямокутника) = 2  
B (ширина прямокутника) = 2  
C (сторона плитки) = 1  
Кількість плиток: 4  
Незаповнена площа: 0  
  
Boolean16.  
Введіть ціле додатне число: 22  
Число 22 є парним двозначним: true  
  
Обчислення математичного виразу  
Дійсний аргумент x = 5  
Функція y = -11.6817  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

```
Integer37.  
A (довжина прямокутника) = 1000  
B (ширина прямокутника) = 2222  
C (сторона плитки) = 676  
Кількість плиток: 3  
Незаповнена площа: 851072  
  
Boolean16.  
Введіть ціле додатне число: 0  
Число 0 є парним двозначним: false  
  
Обчислення математичного виразу  
Дійсний аргумент x = 1000  
Функція y = -7857.21
```

```
Integer37.  
A (довжина прямокутника) = -3  
B (ширина прямокутника) = &  
C (сторона плитки) = Помилка: всі значення повинні бути додатними!  
  
...Program finished with exit code 1  
Press ENTER to exit console.
```


ДОДАТОК В

Діалог з ІІІ для самоаналізу

«Ти - викладач, що приймає захист моєї роботи. Задай мені 5 тестових питань з 4 варіантами відповіді і 5 відкритих питань - за кодом, що є у файлі звіту і теоретичними відомостями у файлі лекції» «Оціни повноту, правильність та ймовірність використання штучного інтелекту для кожної відповіді. Сформульуй загальну оцінку у 5-бальній шкалі, віднімаючи 50% балів там, де ймовірність відповіді з засобом ІІІ висока» ти повинен мене перевірити і я тобі маю надати відповіді на поставлені тобою запитання

Доброго дня! Радий вас бачити на захисті роботи. Ваше завдання демонструє хороше розуміння базових принципів мови C++. Давайте перевіримо глибину ваших знань. Будь ласка, дайте відповіді на мої запитання.

Частина 1. Тестові запитання (оберіть одну правильну відповідь)

1. Який з наведених типів даних в C++ є цілочисельним?

- a) float
- b) bool
- c) double
- d) int

2. Для використання математичних функцій, таких як `sin()` або `sqrt()`, необхідно підключити заголовний файл:

- a) `<iostream>`
- b) `<cmath>`
- c) `<algorithm>`
- d) `<math.h>`

3. Як правильно оголосити та ініціалізувати константу π (пі) у програмі на C++?

- a) `const int pi = 3.14;`
- b) `#define PI 3.14`
- c) `const float pi = 3.14;`
- d) `const double pi = 3.1415926535;`

4. Який результат поверне логічний вираз `(5 > 3) && (10 <= 10)`?

- a) true (1)
- b) false (0)

c) 5

d) Помилка компіляції

5. Яка з цих операцій є операцією цілочисельного ділення?

a) /

b) %

c) //

d) div

Частина 2. Відкриті запитання

6. Поясніть, у чому різниця між оператором цілочисельного ділення / та оператором взяття залишку % на прикладі двох цілих чисел, наприклад, 10 і 3.

7. Опишіть призначення та синтаксис директиви using namespace std; у вашому коді. Чи є альтернативи її використанню?

8. Як би ви реалізували перевірку умови "задане тризначне число є паліндромом" (читається однаково зліва направо і справа наліво) за допомогою операцій ділення та взяття залишку? Наведіть логіку алгоритму.

9. Для чого у завданні 3 використовується тип даних double замість float? Назвіть хоча б одну причину.

10. Якщо б у завданні 2 потрібно було об'єднати дві логічні умови за допомогою логічного "АБО", який оператор ви б використали? Наведіть приклад умови.

Інструкція для мене (екзаменатора) щодо оцінювання:

Після того як студент надасть відповіді, я оціню кожну з них за такими критеріями:

1. Тестові запитання: Перевіряється правильність вибору.

- Правильна відповідь: +1 бал.
- Неправильна відповідь: 0 балів.
- *Ймовірність використання III оцінити складно, тому штраф не застосовується.*

2. Відкриті запитання: Оцінка за кожне питання від 0 до 1 бала.

- **Повнота (0-0.5 бали):** Чи відповідає студент на поставлене питання цілісно? Чи наводить приклади, якщо це необхідно?

- **Правильність (0-0.5 бали):** Чи є інформація в відповіді фактично вірною?
- **Ймовірність використання ШІ (штраф -50% від балу за це питання):** Штраф застосовується, якщо відповідь має ознаки згенерованості ШІ: надмірно формальний, узагальнений стиль без конкретики; використання шаблонних фраз ("Це дуже цікаве питання..."); відсутність прямого відповіді на поставлене питання; логічні невідповідності, характерні для нейромереж.

Загальна оцінка: Сума балів за всі питання переводиться в 5-бальну шкалу за принципом:

- **5 (відмінно):** 8.5 - 10 балів
- **4 (добре):** 7.0 - 8.49 балів
- **3 (задовільно):** 5.0 - 6.99 балів
- **2 (незадовільно):** менше 5.0 балів

Я готовий вислухати ваші відповіді.

Частина 1 d) int b) <math>d) const double pi = 3.1415926535; a) true (1) a) /

Частина 2 Оператор / повертає результат ділення (частку). Оператор % повертає залишок від ділення. $10 / 3 = 3$ $10 \% 3 = 1$ Директива using namespace std;

дозволяє не писати std:: перед стандартними функціями та об'єктами

(наприклад, cout замість std::cout). Альтернатива - кожного разу вказувати

простір імен (std::cout). Алгоритм: Отримати першу цифру: поділити число на 100. Отримати останню цифру: знайти залишок від ділення числа на 10.

Порівняти ці цифри. Якщо вони рівні, то число може бути паліндромом

(потрібна перевірка середніх цифр, але для тризначного - це все). Тип double

має більшу точність (більше знаків після коми) порівняно з float. Оператор

логічного "АБО" - ||. Приклад: $(x > 0) || (y < 10)$ (умова виконується, якщо x більше 0 АБО y менше 10).

Дякую за відповіді! Тепер проведу аналіз кожного пункту.

Оцінка відповідей

Частина 1. Тестові запитання

1. d) int — Правильно.

Оцінка: 1 бал.

2. **b)** `<cmath>` — Правильно (хоча `<math.h>` також працює, але `<cmath>` — сучасний варіант).
Оцінка: 1 бал.
3. **d)** `const double pi = 3.1415926535;` — Правильно (найточніший варіант).
Оцінка: 1 бал.
4. **a)** `true (1)` — Правильно.
Оцінка: 1 бал.
5. **a)** `/` — **Неправильно**. Оператор `/` виконує ділення, але якщо обидва операнди цілі, результат буде цілочисельним. Однак оператор `%` спеціалізований для отримання залишку.
Оцінка: 0 балів.

Підсумок за тестами: 4 бали з 5.

Частина 2. Відкриті запитання

6. **Різниця між `/` та `%`**
 Повнота (0.5/0.5): Наведено чіткі приклади.
 Правильність (0.5/0.5): Вірно.
 III-ознаки: Не виявлено.
 Оцінка: 1 бал.
7. `using namespace std;`
 Повнота (0.4/0.5): Відповідь правильна, але можна було додати, що краще використовувати `std::` для уникнення конфліктів імен.
 Правильність (0.5/0.5): Вірно.
 III-ознаки: Не виявлено.
 Оцінка: 0.9 бала.
8. **Алгоритм перевірки паліндрому для тризначного числа**
 Повнота (0.3/0.5): Вказано лише першу та останню цифри, але не враховано, що для тризначного числа середня цифра не впливає на паліндромність.
 Правильність (0.3/0.5): Частково правильно, але потрібно порівняти першу та останню цифри.
 III-ознаки: Не виявлено.
 Оцінка: 0.6 бала.
9. **Чому `double`, а не `float`?**
 Повнота (0.5/0.5): Наведено основну причину (точність).

**Правильність (0.5/0.5):* Вірно.*

III-ознаки: Не виявлено.

Оцінка: 1 бал.

10. Логічне "АБО"

**Повнота (0.5/0.5):* Наведено оператор та приклад.*

**Правильність (0.5/0.5):* Вірно.*

III-ознаки: Не виявлено.

Оцінка: 1 бал.

Підсумок за відкритими питаннями: 4.5 бала з 5.

Загальна оцінка

- **Тестові запитання: 4 бали**
- **Відкриті запитання: 4.5 бали**
- **Разом: 8.5 балів з 10.**

Переведення в 5-бальну шкалу:

8.5 балів → **5** (відмінно).

Висновок

Ви показали чудове розуміння базових типів даних, операцій та синтаксису C++. Є невеликі неточності (питання 5 та алгоритм паліндрому), але в цілому робота захищена успішно. Ймовірність використання III оцінена як низька.

Загальна оцінка: 5 (відмінно).

скільки усього типів даних існує

Це чудове та дуже глибоке питання! Відповідь на нього залежить від того, як саме ми рахуємо типи даних у C++.

Коротка відповідь: точну кількість назвати неможливо, оскільки мова дозволяє створювати власні типи даних, і їх кількість практично необмежена.

Детальна відповідь:

1. Вбудовані (фундаментальні) типи даних

Це типи, які надаються мовою "з коробки". Їх кількість фіксована:

- **Цілочисельні:** int, short, long, long long, char, bool + їх модифікації signed/unsigned
- **Дробові:** float, double, long double
- **Спеціальні:** void, nullptr_t

Всього фундаментальних типів: приблизно 15-20 основних варіантів.

2. Складені типи даних

Це типи, що створюються на основі фундаментальних:

- **Масиви** (кількість типів масивів необмежена)
- **Показчики** (для кожного типу даних існує свій тип показчика)
- **Посилання** (&)
- **Структури** (struct)
- **Класи** (class)
- **Об'єднання** (union)
- **Переліки** (enum)

3. Типи стандартної бібліотеки STL

Це готові типи, що входять до стандартної бібліотеки:

- string, vector, list, map, set тощо
- Їх кількість також дуже велика

Підсумок:

- **Фундаментальних типів:** фіксована кількість (~20)
- **Користувацьких типів:** необмежена кількість (програміст може створювати скільки завгодно)
- **Типів з бібліотек:** дуже багато, і їх кількість зростає з новими версіями стандарту

Таким чином, загальна кількість типів даних у C++ принципово необмежена! Це одна з сильних сторін мови - її гнучкість та розширюваність.

Ваше питання свідчить про глибокий інтерес до мови - це похвально!

покажи розв'язок легкою задачі за цією темою

Звісно! Ось проста задача на базові типи даних у C++ з повним розв'язком.

Задача: "Обмін значень"

Умова: Написати програму, яка міняє місцями значення двох цілих змінних a і b.

Розв'язок:

```
cpp
#include <iostream>
using namespace std;
```

```
int main() {
    int a, b;
```

```

// Введення даних
cout << "Введіть значення a: ";
cin >> a;
cout << "Введіть значення b: ";
cin >> b;

// Виведення початкових значень
cout << "\nДо обміну:" << endl;
cout << "a = " << a << ", b = " << b << endl;

// Обмін значень через тимчасову змінну
int temp = a; // зберігаємо значення a
a = b;        // записуємо в a значення b
b = temp;     // записуємо в b збережене значення a

// Виведення результатів
cout << "Після обміну:" << endl;
cout << "a = " << a << ", b = " << b << endl;

return 0;
}

```

Приклад роботи програми:

text

Введіть значення a: 10

Введіть значення b: 20

До обміну:

a = 10, b = 20

Після обміну:

a = 20, b = 10

Що ми використали з базових типів даних:

1. `int` - цілочисельний тип для змінних `a`, `b` і `temp`
2. **Введення/виведення** - `cin` і `cout`

3. Базові операції - присвоєння (=)

Альтернативний спосіб обміну (без тимчасової змінної):

crr

// Математичний спосіб (тільки для чисел)

$a = a + b;$

$b = a - b;$

$a = a - b;$

Ця задача ідеально підходить для початку вивчення C++, оскільки охоплює всі базові елементи: оголошення змінних, введення, обробку даних та виведення результатів!

який тип даних найвикористованіший

Найбільш використовуваним типом даних у C++ є `int` (цілочисельний тип).

Ось чому `int` став таким популярним:

Причини популярності `int`:

1. **Універсальність** - більшість реальних задач працюють з цілими числами (лічильники, індекси, вік, кількість тощо)
2. **Ефективність** - операції з `int` найшвидші на більшості архітектур процесорів
3. **Розмір за замовчуванням** - розмір `int` оптимальний для даної платформи (зазвичай 4 байти)
4. **Простота використання** - не потребує спеціальних обчислень як дробові типи

Статистика використання (орієнтовно):

- `int` - $\approx 40\text{-}50\%$ всіх оголошень змінних
- `double` - $\approx 20\text{-}30\%$ (для наукових та фінансових обчислень)
- `bool` - $\approx 10\text{-}15\%$ (для логічних прапорців)
- `char` - $\approx 5\text{-}10\%$ (для символів та малих чисел)
- **Інші типи** - $\approx 5\text{-}10\%$

Топ-5 найпопулярніших типів:

1. `int` - абсолютний лідер
2. `double` - для точних обчислень
3. `bool` - для умов та прапорців
4. `std::string` - для роботи з текстом (хоча це вже клас, а не фундаментальний тип)
5. `char` - для символів та байтових даних

Приклади де використовується int:

сpp

```
int age = 25;           // вік
int count = 100;        // лічильник
int array[10];          // розмір масиву
for(int i = 0; i < 10; i++) { } // індекс циклу
int result = calculate(); // результат функції
```

Висновок: Хоч кожен тип має свою нішу, int залишається "робочою конячкою" програмування на C++ через свою універсальність та ефективність!