**Лабораторна робота №1**

**Розробка програм з лінійною структурою**

***Мета роботи***

* Ознайомитись з середовищем програмування Visual Studio 2010/2019(С++)
* Навчитись створювати консольні додатки, компілювати програму, вико­нувати, налагоджувати, користуватись системою допомогии і підказками.
* Засвоїти основні символи графічних схем, які використовуються у програмах лінійної структури.
* Ознайомитись з етапами розробки програм.
* Навчитися розробляти програми з лінійною структурою, тобто програм, у яких всі оператори виконуються послідовно, один за одним.
* Навчитися формувати звіт з лабораторної роботи.

Завдання 1.1

Варіант 14

А) 

**Формалізація задачі**

1. Вхідні дані: B.
2. Вихідні дані: z.
3. Типи даних. Оскільки значення B, z в синусі/косинусі можуть бути і цілими, і дійсними, доцільно обрати тип *float.*
4. Перевірка правильності. Для перевірки правильності обчислень, слід підготувати декілька контрольних прикладів.

**Словесний алгоритм**

1. Початок.
2. Оголошення змінних: B, z – дійсні.
3. Видати запрошення для введення B.
4. Ввести B.
5. Обчислити z.
6. Вивести на екран значення z.
7. Кінець.

1

1

1

**Лістинг програми**

#include <iostream>

#include <math.h>

using namespace std;

int main()

{ float z2, B;

cout << "\nEnter B >>> "; cin >> B;

B = B \* (3.1415 / 180);

z2 = (1 + sin(2 \* B)) / cos(2 \* B);

cout <<"\nz2 = " << z2;

}

**Результати**

|  |  |  |
| --- | --- | --- |
| № | Програма | Контрольний приклад |
| 1 | Enter B >>>  90  z2 = -1.00009 | B=90  z2 = -1.00009 |
| 2 | Enter B >>>  -45  z2 = 0 | B=-45  z = 0 |
| 3 | Enter B >>>  0  z2 = 1 | B = 0  z = 1 |

Б) 

**Формалізація задачі**

1. Вхідні дані: x.
2. Вихідні дані: z1.
3. Типи даних. Оскільки значення x, z1 в синусі/косинусі можуть бути і цілими, і дійсними, доцільно обрати тип *float.*
4. Перевірка правильності. Для перевірки правильності обчислень, слід підготувати декілька контрольних прикладів.

**Словесний алгоритм**

1. Початок.
2. Оголошення змінних: х, z1– дійсні.
3. Видати запрошення для введення x, z1.
4. Ввести x.
5. обчислити z1.
6. Вивести на екран значення z1.
7. Кінець.

**Лістинг програми**

**Результати**

#include <iostream>

#include <math.h>

using namespace std;

int main()

{

float z1, x;

xlink:

cout << "\nEnter X >>> "; cin >> x;

if (x < 4) {

cout << "X must be > 3";

goto xlink;

}

z1 = (pow(x, 2) + 2 \* x - 3 + (x + 1) \* sqrt(pow(x, 2) - 9)) / (pow(x, 2) - 2 \* x - 3 + (x - 1) \* sqrt(pow(x, 2) - 9));

cout << "\n z1 = " << z1;

|  |  |  |
| --- | --- | --- |
| № | Програма | Контрольний приклад |
| 1 | Enter x >>> 4  z1 = 2.64575 | x = 4  z1 = 2.64575 |
| 2 | Enter x >>> -3  X must be >3 | x = -3  X must be >3 |
| 3 | Enter x >>> 0  X must be >3 | X = 0  X must be >3 |

Завдання 1.2

Варіант 14 Визначення за заданими x та a значень x a , x -a , a/ x ,x/a , x%a ,x+1,a -1;

**Формалізація задачі**

1. Вхідні дані: x, a.
2. Вихідні дані:x, a.
3. Типи даних. Оскільки значення a ,x можуть бути і цілими, і дійсними, доцільно обрати тип *double.*
4. Якщо а або x дорівнює 0 то дані не правильні.
5. Перевірка правильності. Для перевірки правильності обчислень, слід підготувати декілька контрольних прикладів.

**Словесний алгоритм**

1. Початок.
2. Оголошення змінних: a, x.-дійсні.
3. Видати запрошення для введення a, x.
4. Ввести a, x.
5. Якщо a, x рівні 0, то вивести помилку .
6. Обчислити x -a , a/ x ,x/a , x%a ,x+1,a -1;.
7. Вивести на екран значення x -a , a/ x ,x/a , x%a ,x+1,a -1.
8. Кінець.

**Лістинг програми**

#include <iostream>

using namespace std;

int main()

{ double x, a;

cout << "Enter x: ";

cin >> x;

cout << "Enter a: ";

cin >> a;

cout << "x \* a = " << x \* a << endl;

cout << "x - a = " << x - a << endl;

if (x == 0) {

cout << "division by 0 is not possible." << endl;

}

else {

cout << "a / x = " << a / x << endl;

}

if (a == 0) {

cout << "division by 0 is not possible." << endl;

}

else {

cout << "x / a = " << x / a << endl;

}

if (a == 0) {

cout << "division by 0 is not possible." << endl;

}

else {

cout << "a % x = " << int(a) % int(x) << endl;

}

cout << "x + 1 = " << x + 1 << endl;

cout << "a - 1 = " << a - 1 << endl;

}

**Результати**

|  |  |  |
| --- | --- | --- |
| № | Програма | Контрольний приклад |
| 1 | Enter x: 3  Enter a: 4  x \* a = 12  x - a = -1  a / x = 1.33333  x / a = 0.75  a % x = 1  x + 1 = 4  a - 1 = 3 | x = 3  a = 4  x \* a = 12  x - a = -1  a / x = 1.33333  x / a = 0.75  a % x = 1  x + 1 = 4  a - 1 = 3 |
| 2 | Enter x: 0  Enter a: 0  x \* a = 0  x - a = 0  division by 0 is not possible.  division by 0 is not possible.  division by 0 is not possible.  x + 1 = 1  a - 1 = -1 | a = 0  x = 0  x\* a = 0  x – a = 0  x+1 = 0  a – 1 = -1 |
| 3 | Enter x: -3  Enter a: -2  x \* a = 6  x - a = -1  a / x = 0.666667  x / a = 1.5  a % x = -2  x + 1 = -2  a - 1 = -3 | X = -3  A = -2  x \* a = 6  x - a = -1  a / x = 0.666667  x / a = 1.5  a % x = -2  x + 1 = -2  a - 1 = -3 |



Контрольні питання

1. Які середовища програмування мовою С ++ ви знаєте?   
   С++ середовища програмування: Visual Studio, Code::Blocks, CLion, Dev C++, Xcode.
2. Охарактеризуйте коротко переваги і недоліки середовища програмування Visual Studio (C++)?   
   Недоліки - великий ресурсомісткий, важкий для початківців.
3. У чому різниця між інтерпретатором і компілятором?   
   Компілятор перетворює вихідний код в машинний код, інтерпретатор виконує код рядок за рядком.
4. Як відбувається компіляція вихідного коду програми?   
   Вихідний код перетворюється у виконуваний файл компілятором.
5. Для чого існує компонувач?   
   Компонувач використовується для об'єднання різних об'єктних файлів у виконуваний файл.
6. Які види бібліотек ви знаєте? В чому різниця між ними?   
   Види бібліотек: стандартні, зовнішні; різниця - стандартні є частиною мови, зовнішні створюються користувачами.
7. Наведіть основні типи даних, наведіть приклади.   
   Основні типи даних: int, float, char; приклади - int age = 25, float price = 10.99, char grade = 'A'.
8. Який порядок виконання операцій у складних виразах у програмах?   
   Операції виконуються в порядку: дужки, множення/ділення, додавання/віднімання.
9. Яким чином здійснюються перетворення типів у програмах мовою С++?   
   Перетворення типів: явне (int to float), неявне (int to double); правила - можливі втрати даних.
10. Яка головна функція у програмах мовою С++?   
    int main() {...}.
11. Яка послідовність дій при розробці програми?   
    Aналіз, проектування, кодування, тестування, вдосконалення.
12. Наведіть основні символи для виконання лінійних схем виконання програм і поясніть їх.   
    Основні символи: стрілки (->), круглі дужки (), фігурні дужки {}, квадратні дужки [].
13. Які ви знаєте функції для введення інформації з клавіатури  і виведення   
    на екран?  
    cin для вводу, cout для виведення.
14. Як можна задати формат виведення інформації на екран?   
    Формат виведення: cout << "Число: " << num << endl; для виведення числа "num" з новим рядком.
15. Наведіть і поясніть різні види операторів присвоєння.   
    **x = 10;** - присвоєння значення 10 змінній **x**.

**x += 5;** - додавання 5 до поточного значення **x** і присвоєння результату змінній **x**.  
**x -= 3;** - віднімання 3 від поточного значення **x** і присвоєння результату змінній **x**.  
**x \*= 2;** - множення поточного значення **x** на 2 і присвоєння результату змінній **x**.  
**x /= 4;** - ділення поточного значення **x** на 4 і присвоєння результату змінній

**Висновки:**

1. Засвоєно основні правила роботи  у середовищі Visual Studio: порядок створення проектів, використання підказок, способи налагодження.
2. Засвоєно використання таких функцій:
   * main() – головна функція програми;
   * cout << –виведення на екран;
   * cin >> – введення з клавіатури ;
3. Отримано практичні навички із застосування таких базових понять: оператори присвоєння; основні типи даних; порядок математичних операцій у виразі; оператор препроцесором обробки #include.
4. Засвоєно розробку схем лінійних програм.

