

Алгоритми та структури даних
Модуль 1: Основи програмування
та структури даних
Лекція 1. Основи програмування
на C++

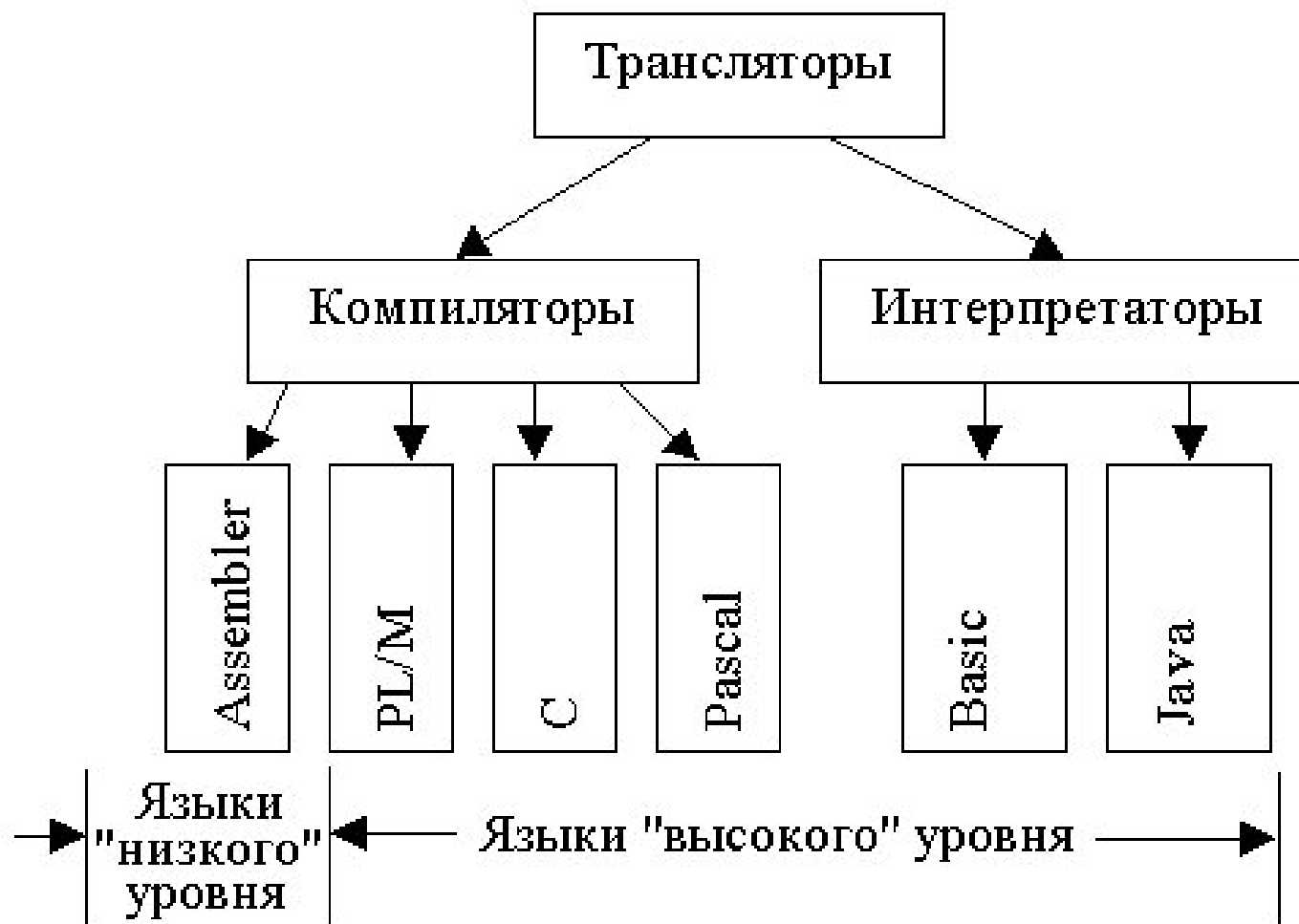
Кудін Олексій Володимирович
avk256@gmail.com









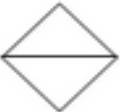



План

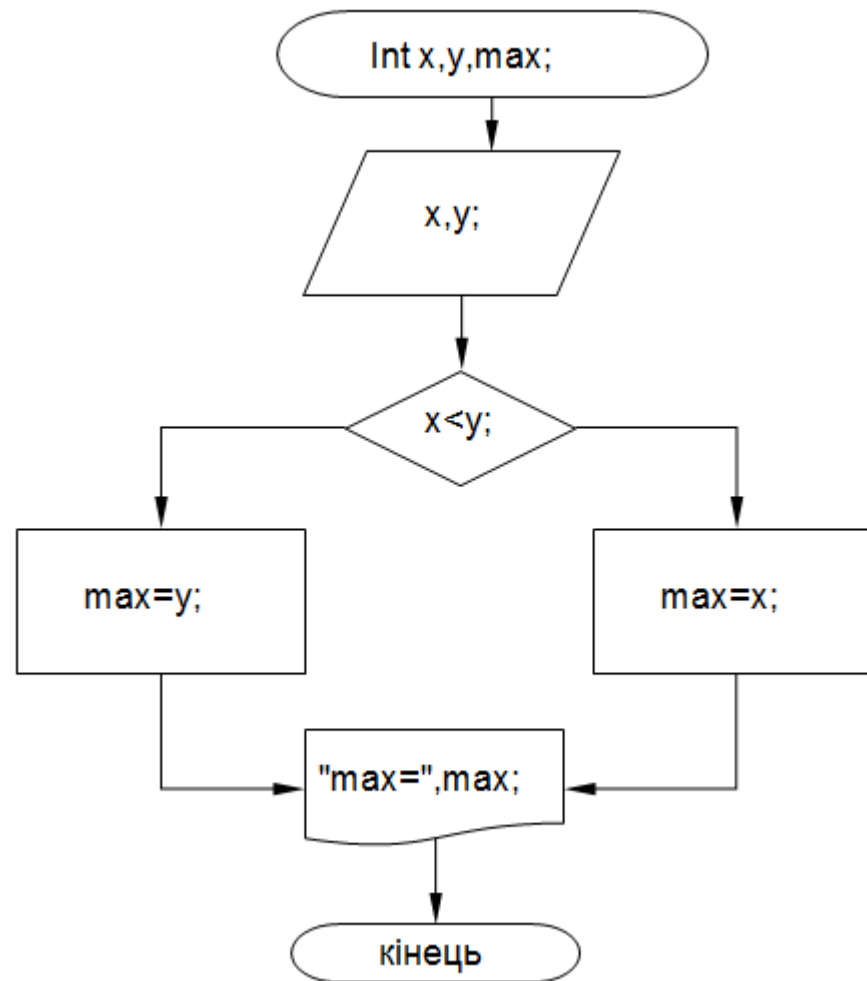
1. Класифікація мов програмування.
2. Основні алгоритмічні конструкції.
3. Етапи розв'язання обчислювальних задач на комп'ютері.
4. Алфавіт мови програмування.
5. Структура програми.
6. Умовні оператори.
7. Оператори циклу.
8. Оператори передачі управління.

- **Імперативні** мови програмування орієнтовані на оператори. Обчислення в них представляються як послідовність операторів (команд) (Fortran, Algo, Cobol, C\C++).
- **Функціональні** мови програмування задають процес обчислень як послідовність функцій (Lisp, Haskell).
- **Логічні** мови програмування описують обчислення за допомогою формальної логіки (Prolog).
- **Об'єктно-орієнтовані** мови програмування – обчислення реалізуються у вигляді сукупності об'єктів (C#, Java).
- Мови програмування, що **інтерпретуються**.
- Мови програмування, що **компілюються**.
- **Скриптові мови програмування** (Perl, JavaScript, PHP)



- Способи запису алгоритмів:
 1. Словесний опис.
 2. Графічний опис (блок-схема).
 3. Псевдокод.

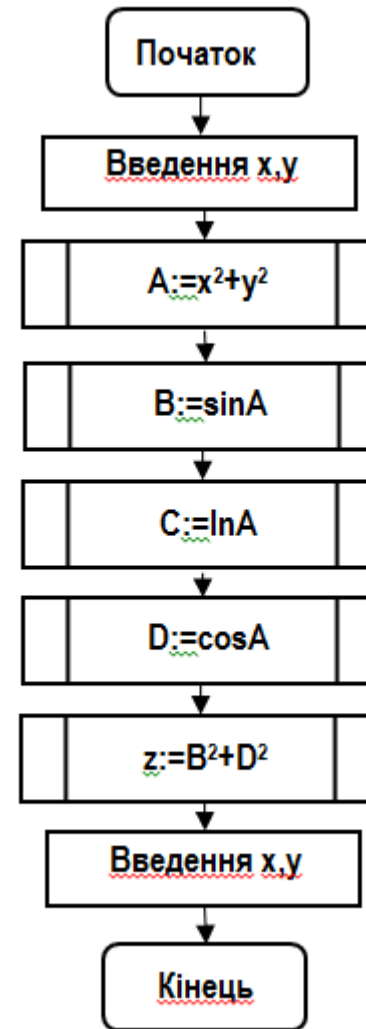
Геометричне подання	Призначення оператора	Геометричне подання	Призначення оператора
	Процес (арифметичний оператор)		Розв'язок (умовний оператор)
	Дані (оператори введення-виведення)		Початок-кінець (оператор зупинки або початок блок-схеми)
	Документ (друк на бумагу)		Монітор (виведення на монітор)
	Сортування		Збереження даних
	Пам'ять з послідовним доступом		Пам'ять з прямим доступом



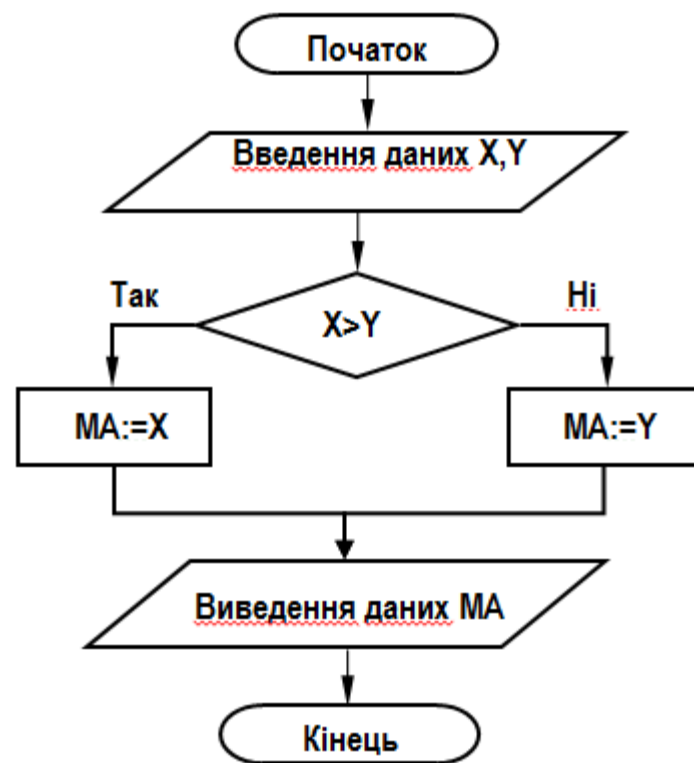
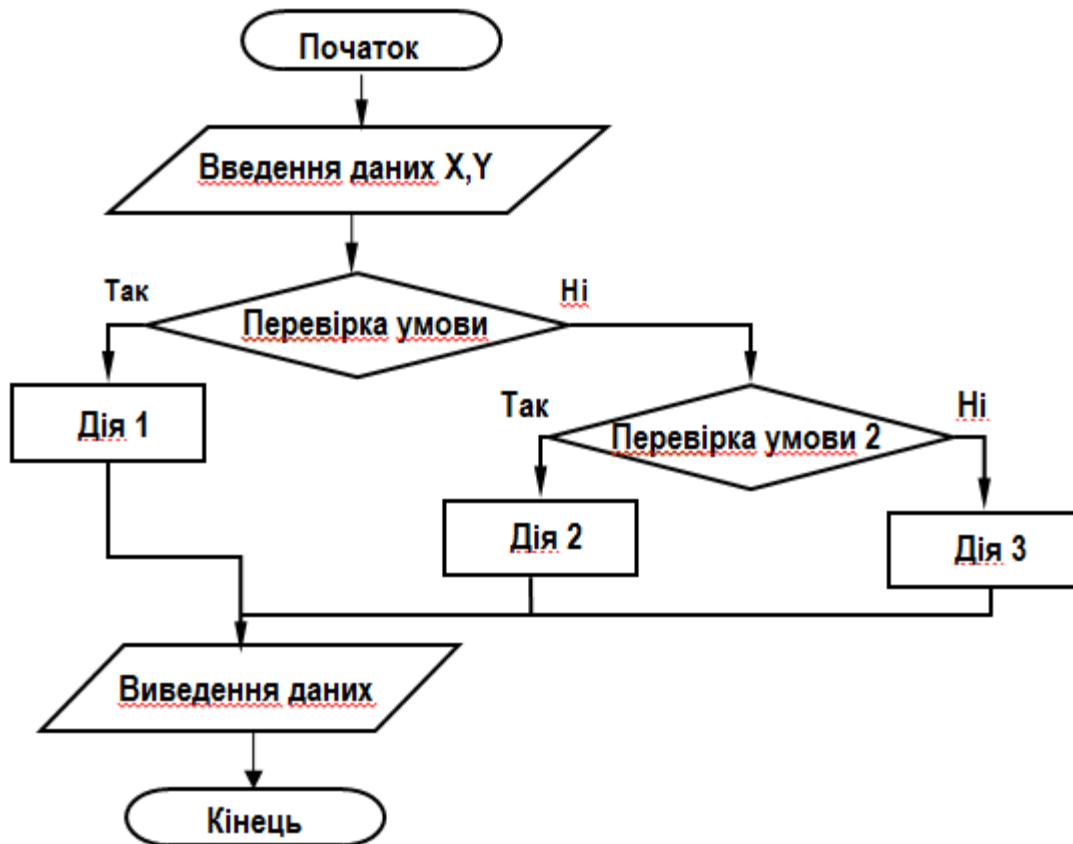
- **Псевдокод** – це неформальний запис алгоритму, що використовує структуру поширених мов програмування, але нехтує деталями коду, неістотними для розуміння алгоритму

```
Repeat
  Compute  $EMA(T)$ 
  If no position opened
    If  $EMA(T) \geq P$ 
      If trend is going up
        Open a long position
      Else if trend is going down
        Open a short position
    Else if any position is opened
      If  $EMA(\neg T) \geq Q$ 
        Close position
  If end of market
    Close all opened position
Until market close
```

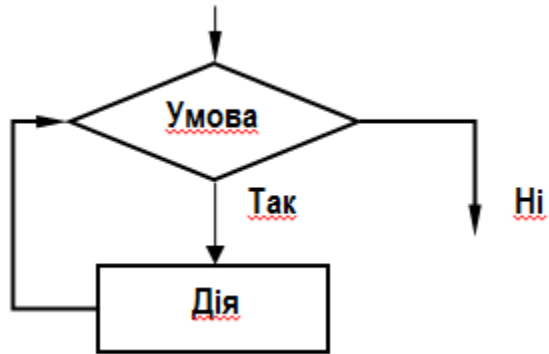

■ Лінійний алгоритм



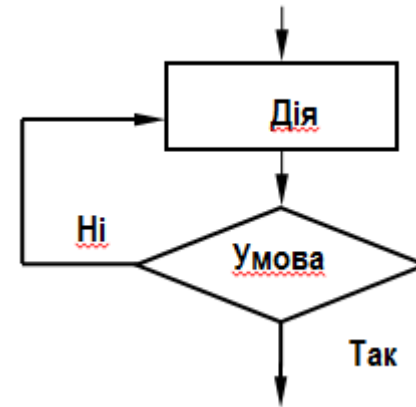
■ Розгалужений алгоритм



■ Цикл з передумовою



Цикл з постумовою



Цикл з параметром

Цикл з параметром



- **Алфавіт** мови програмування – це скінченний набір символів. За допомогою цих символів можуть бути записані ідентифікатори, вирази та оператори мови.
- Із символів алфавіту складаються лексичні одиниці мови, або **лексеми** – мінімальні значущі одиниці в текстах програм. Множина всіх допустимих лексем називається **словником** мови програмування.
- **Лексеми спеціальних символів**, окрім спеціальних символів з алфавіту мови, містять ще складені спеціальні символи, що сприймаються компілятором як єдине ціле (`<=`, `>=`, `+=`, `*=`, `++`, `==`, `--` і т.д.).
- **Зарезервовані (ключові) слова** мають строго визначений зміст. Їх призначення не може змінюватися. Зарезервовані слова використовуються для позначення алгоритмічних конструкцій (циклів, умов, операторів), типів змінних тощо.

- **Ідентифікатор** – це ім'я, значення якого може варіюватися від програми до програми або навіть у межах однієї програми. У мові C\C++ розрізняють стандартні ідентифікатори та ідентифікатори користувача.
- **Директива препроцесора** – це рядок, що починається символом #. Директиви препроцесора визначають режими компіляції і можуть істотно впливати на зміст згенерованого компілятором машинного коду.

- Записана мовами C\C++ програма складається з трьох частин:
- директив препроцесора;
- декларативної (оголошення ідентифікаторів, що використовуються у програмі);
- функцій (іменованих частин програми, які містять запис дій, що виконуються).

Приклад 1. Введення двох чисел та обчислення їх суми.

```
// example 1.cpp ввести два числа та визначити їх суму
#include "stdafx.h" // Включення заголовних файлів
#include <iostream>
using namespace std; // Визначення простору імен
int a; // Оголошення змінних
float b;

void input() // Визначення функції
{
    cout << "Enter int a, float b" << endl; // Вивід строки з запрошенням
    cin >> a >> b; // Ввід даних
}
```

ВВОДУ ДАНИХ


```
int main()
{
    cout << "Hello, world" << endl;           // Вивести повідомлення

    input();                                   // Виклик функції
    cout << "a+b= " << a + b << endl;       // Вивід строки з результатом
обчислень
    return 0;
}
```

Тип даних визначає:

- множину допустимих значень, яких може набувати змінна або константа зазначеного типу;
- множину допустимих операцій, що застосовуються до даних певного типу;
- спосіб зображення даних у пам'яті комп'ютера.

Табл. 1 – Цілочислові типи

Ідентифікатор типу	Кількість байтів	Діапазон значень
char	1	-128..+127 (2^7-1)
unsigned char	1	0..255 (2^8-1)
short	2	-32768..32768 ($2^{15}-1$)
unsigned short	2	0..65535 ($2^{16}-1$)
<u>int</u>	4	-2147483648.. 2147483647 ($2^{31}-1$)
unsigned <u>int</u>	4	0..4294967295 ($2^{32}-1$)
long	4	-2147483648.. 2147483647 ($2^{31}-1$)
unsigned long	4	0..4294967295 ($2^{32}-1$)

Табл. 2 – Арифметичні операції та операції порівняння

Знак операції	Зміст операції
+	Додавання
-	Віднімання
*	Множення
/	Визначення цілої частини від ділення
%	Визначення остачі від ділення
- (<u>унарний</u>)	Зміна знаку числа
++	<u>Префіксний</u> та <u>постфіксний інкремент</u> (збільшення на 1)
--	<u>Префіксний</u> та <u>постфіксний декремент</u> (зменшення на 1)
<<	Зсув бітів ліворуч ($2 \ll 1 = 4$, $1 \ll 5 = 32$)
>>	Зсув бітів праворуч ($2 \gg 1 = 1$, $8 \gg 2 = 2$)
==	Дорівнює
!=	Не дорівнює
>	Більше
<	Менше
>=	Більше дорівнює (не менше)
<=	Менше дорівнює (не більше)

Табл. 3 – Дійсні типи даних

Назва	Кількість байтів	Найменше за модулем число	за модулем	Найбільше за модулем число
float	4	$3,4 \times 10^{-38}$		$3,4 \times 10^{38}$
double	8	$1,7 \times 10^{-308}$		$1,7 \times 10^{308}$
long double	10	$3,4 \times 10^{-4932}$		$1,1 \times 10^{4932}$

Табл. 4 – Булеві операції

a	b	a&&b	a b	!a
false	false	false	false	true
false	true	false	true	true
true	false	false	true	false
true	true	true	true	false

■ Оператори if та if-else

```
if (вираз)  
    оператор
```

```
if (вираз)  
    оператор 1  
else  
    оператор 2
```

■ Оператор while

```
while (выраз)  
    оператор
```

■ Оператор do

```
do  
    оператор  
while (выраз)
```

■ Оператор for

for (вираз 1; вираз 2; вираз 3)
оператор

- Приклад. Програма обчислення суми цілих чисел від 0 до 10.

```
sum = 0;
```

```
for (i = 1; i <= 10; ++i)
```

```
    sum += i;
```


Оператори передачі управління

Оператори break і continue

Оператор switch

switch (вираз)

case вираз 1: оператор 1; break;

case вираз 2: оператор 2; break;

...

default:

оператор за замовченням;

Резюме (що встигли за лекцію)

- Зробили огляд видів мов програмування
- Розглянули способи запису алгоритмів
- Розглянули структуру програми C++
- Навели основні типи даних
- Розглянули оператори розгалуження та цикли