



# Типи даних мови С (C++)

Лектор:

к.т.н., доцент кафедри прикладної математики

Рижа Ірина Андріївна

# Про що ця лекція

---

- Розглянемо поняття змінної та константи.
- Опишемо базові типи даних у С (C++) та їх представлення в пам'яті.

# Змінні та константи

---

## Змінна

- комірка з області оперативної пам'яті, де зберігається вхідна, допоміжна чи результуюча інформація (дані), яка може бути зміненою.
- Використання змінних, а не їх значень, дозволяє один і той самий програмний код використати для розв'язання задач із різними вхідними значеннями.

## Константа

- область оперативної пам'яті, де зберігається інформація, що не підлягає зміні.

# Ідентифікатори

## Ідентифікатор

– це ім'я змінної, функції, константи в програмуванні.

## Правила задання ідентифікатора.

1. Ідентифікатори є послідовністю символів абетки мови програмування, тобто не можуть містити інших символів, ніж латинські літери (малі та великі), символ підкреслення і цифри.
2. Ідентифікатор не може починатися з цифри, лише з літери чи підкреслення.
3. Малі та великі символи в ідентифікаторах розрізняються.
4. Як ідентифікатори не можна використовувати ключові слова.
5. Одне й те саме ім'я не можна оголошувати двічі й більше разів в межах одного блоку.
6. Назви змінних бажано давати змістовно, щоб ідентифікатор вказував на суть змінної та для чого вона оголошена.
7. Довжина ідентифікатора, тобто кількість використаних символів теоретично не обмежується

# Ключові слова

## Ключові слова

– слова, зарезервовані для запису інструкцій, що містять вказівки комп'ютеру щодо способу реалізації завдання.

- Всі зарезервовані слова пишуть з маленької літери, а от константи, які означені програмним середовищем, записані великими літерами.
- До ключових слів належать такі, які редактор підсвічує блакитним кольором, а також ім'я головної функції проекту, тобто `main`.
- Це слова, що задають стандартні типи даних, назви операторів умовного, безумовного переходу, оператор вибору, оператор циклу, модифікатори та специфікатори.

# Типи даних

---

## Основні принципи збереження інформації в комп'ютері:

- де зберігається інформація;
- яке значення там зберігається;
- який тип інформації, що зберігається.

Мова програмування C **не підтримує** неявного задання типів даних чи задання за замовчуванням, а тому необхідно обов'язково перед першим використанням даного його оголосити.

# Оголошення змінних

---

Усі змінні величини у C - програмі повинні бути описані.

- Опис змінної містить специфікацію типу, ім'я змінної та ряд необов'язкових модифікаторів, які задають особливості її внутрішнього представлення або клас зберігання.
- При описі змінної є можливість її ініціалізації (присвоєння початкового значення).
- Оголошувати дані можна в тому місці коду програми, де в них виникла потреба.

# Оголошення змінних

Приклад оголошення змінної на мові C:

тип\_даного ім'я\_змінної;

або

тип\_даного список\_змінних;

або

тип\_даного ім'я\_змінної = значення;

*При оголошенні змінної чи змінних відбувається їх означення, тобто виділення під ці змінні належного розміру пам'яті.*



# Класифікація типів даних

Класифікація типів даних може бути різною.

1. «Порожній» -void;
2. Скалярний:
  - арифметичні типи даних (цілочисельні і дійсні);
  - перерахування;
  - вказівники;
  - посилання (у C++);
3. Тип-функція;
4. Агрегований (складений):
  - масиви;
  - структури;
  - класи (у C++).

# Класифікація типів даних

---

## 1. Основні:

- числові;
- символьні;
- логічні;
- void (порожній тип);

## 2. Похідні:

- вказівники;
- посилання;
- масиви;
- функції;
- структури;
- об'єднання;
- класи (у C++).

# Дані цілого типу

## Дані цілого типу

— дані (інформація), що носять цілочисельний характер.

**Базовим цілочисельним типом** є тип, що задається ключовим словом `int`.

- Розмір пам'яті для цього типу визначається програмним середовищем, а MS Visual Studio виділяє під цей тип 4 байти (32 біти).
- При цьому найлівіший біт області пам'яті використовується для представлення знаку змінної (0 — додатній, 1 — від'ємний). Від'ємні числа зберігаються у додатковому коді.
- Значення чисел цього типу лежать у діапазоні

від  $-2147483648$  ( $-2^{31}$ ) до  $+2147483647$  ( $2^{31} - 1$ ).

```
int number = 22;
```

```
int a, b = 2, c;
```

# Дані цілого типу

## Модифікатор типу `unsigned`

- Змінює формат та діапазон представлення даних, але не впливає на розміри виділених областей пам'яті.
- Вказує, що змінна приймає лише невід'ємні значення.
- При цьому найлівіший біт області пам'яті, як і всі інші біти області пам'яті – використовуються для представлення цифрової частини.

`unsigned int score;`

- Мова С дозволяє при цьому не писати слова `int`.
- Діапазон зміни таких цілих чисел становить

від 0 до 4295967295 ( $2^{32}$ ).

# Дані цілого типу

## Короткий цілий тип `short`

- Займає 2 байти (16 бітів) оперативної пам'яті.
- Числові значення містяться в межах

від  $-32768$  ( $-2^{15}$ ) до  $+32767$  ( $2^{15} - 1$ ).

- «Беззнакові» числові значення містяться в межах

від 0 до 65535 ( $2^{16}$ ).

`short` number = -4;

`unsigned short` pos = 1;

# Дані цілого типу

## Довгий цілий тип `long`

- Займає 4 байти (32 біти) оперативної пам'яті.
- Числові значення містяться в межах (як у 4-байтного цілого числа)

від  $-2147483648$  ( $-2^{31}$ ) до  $+2147483647$  ( $2^{31} - 1$ ).

- «Беззнакові» числові значення містяться в межах

від 0 до  $4295967295$  ( $2^{32}$ ).

```
long int la = 5L;
```

```
long bl;
```

```
unsigned long ba=3234567;
```

# Дані цілого типу

Тип `short int` є наймолодшим, а `unsigned long` найстаршим, тобто ранжування типів є наступним:

`short int` → `unsigned short int` → `int` → `unsigned int` → `long` → `unsigned long`

- Інформацію про можливі значення даних цілочисельного типу можна отримати із заголовного файлу `climits`.

## Зауваження

Характерною особливістю цілих типів даних є «**закільцьованість**» множини значень, тобто після максимального можливого значення слідує мінімальне.

# Дані символьного типу

Символьним тип призначений для роботи з окремими символами.

- Є підмножиною даних цілого типу, над яким можна виконувати в тому числі й арифметичні дії.
- Такі дані оголошують за допомогою ключового слова `char`.
- Константа типу `char` записується як символ, обмежений апострофами, наприклад `'H'`, `'#'`.
- Такі дані беріають код символу з кодової таблиці символів ASCII.
- Стандартний символьний тип займає 1 байт (8 бітів) і є знаковим.



# Дані символьного типу

- Цілочисельний діапазон даних символьного типу є
  - від  $-128$  ( $-2^7$ ) до  $127$  ( $2^7 - 1$ ),

хоча значення кодів символів є від 0 до 255.

Отже, якщо буде оголошено дане символьного типу і йому нададуть значення символу, код якого становить **129**, то при виведенні на екран цього даного відобразиться відповідний символ, а при виведенні цілочисельного еквіваленту коду з'явиться число – **127**.

- Для того, щоб цілочисельне значення відповідало значенню коду символу використовують «беззнаковий» символьний тип:

`unsigned char` `cm = 'A'` , `sm = 217`, `smb`;

Старшинство типів є наступним:

`char` → `unsigned char` → `short int` → ...

# Дані символьного типу

Є група символьних констант, які інтерпретуються компілятором особливим чином:

- `\n` – символ завершення рядка і переведення курсору на наступний рядок на екрані монітора;
- `\t` – символ табуляції, тобто пропущення 8 позицій;
- `\r` – переведення курсору на початок поточного рядка;
- `\b` – повернення на одну позицію вліво;
- `\a` – надання звукового сигналу;
- `\\` – виведення лівої похилої як символу;
- `\"` – виведення на екран лапок;
- `\'` – виведення на екран апострофа, як одного символу;
- `\0` – задання нуль-символа чи нуль-термінатора, символу завершення рядка символів на C/C++.

# Логічний тип даних

- Вважається, що будь-який ненульовий результат є істинним, а кожен нуль є хибним значенням.
- Окремий тип даних логічного типу задається з допомогою ключового слова `bool`, що є скороченням `boolean`.
- Дані цього типу набувають лише двох значень `false` та `true` (хиба та істина), причому `false < true`.
- При відображенні значення даного логічного типу буде виводитися 0 при значенні `false` і 1 при значенні `true`.
- Такі дані займають лише один двійковий розряд, але для їх збереження виділяється пам'ять розміром в один байт

За старшинством типів цей тип є найнижчим (найменшим), тобто

`bool` → `char` → `unsigned char` → `short int` → ...

# Перелічуваний тип даних (enum)

## Змінна перелічуваного типу

– змінна, котра може приймати значення з деякого списку значень, сформованого користувачем.

- Дані перелічуваних типів внутрішньо зображаються та опрацьовуються як дані цілого типу.

```
enum season {winter, spring, summer, autumn};
```

//цілочисельні еквіваленти за замовчуванням лежать в межах від 0 до 3

```
season S = summer;
```

```
enum month {January = 1, February, March, April, May, June, July, August, September, October,  
November, December};
```

//цілочисельні еквіваленти лежать в межах від 1 до 12

- Не можна змінній перелічуваного типу присвоювати значення, що не зазначені в переліку.

# Дійсні типи даних

## Дійсні типи даних

використовуються для роботи з дійсними числами, тобто даними, що у своєму записі містять дробову частину.

- Дані дійсних типів є старшими відносно даних цілих типів.

Внутрішнє представлення дійсних типів є у **формі з плаваючою крапкою**.

- При цьому мантиса завжди нормалізована, тому старшу одиницю не зберігають в пам'яті.
- Для того, щоб не виділяти окремого біта під знак порядку та спростити реалізацію операцій вирівнювання порядків при виконанні арифметичних операцій над ними, величина порядку зберігається у так званій «зміщеній» формі так, щоб порядок завжди був невід'ємним.

# Дійсні типи даних

Базовим дійсним типом в MS Visual Studio є тип `double`, який займає 8 байт (64 біти).

- Абсолютне значення числа цього типу є в діапазоні

від  $1.7 \cdot 10^{-308}$  до  $1.7 \cdot 10^{+308}$ .

знак	порядок		мантиса	
63	62	52	51	0

# Дійсні типи даних

Дане типу `float` займає 4 байти (32 біти).

- Абсолютне значення числа цього типу є в діапазоні

від  $3.4 \cdot 10^{-38}$  до  $3.4 \cdot 10^{+38}$ .

знак	порядок		мантиса	
31	30	23	22	0

# Дійсні типи даних

Для чисел з великим числом десяткових розрядів використовують тип `long double`.

- Кількість розрядів для мантиси та порядку, а також розмір пам'яті, що її займає `long double`, визначається компілятором.
- Дані цього типу займають в MS Visual Studio 8 байт (64 біти)

Порядок старшинства дійсних типів є таким:

`unsigned long` → `float` → `double` → `long double`



# Дійсні типи даних

Дані дійсного типу подаються у двох формах запису:

- з десятковою крапкою

[знак\_числа]ціла\_частина.дробова\_частина[суфікс]

- з десятковим порядком

мантисаE[знак\_порядку]порядок

суфікс – це необов'язковий елемент запису числа.

Математичний запис числа	Запис числа на мові C (C++)
297.7	297.7
$2.977 \cdot 10^2$	2.977E2 2.977e+2 2977E-1 2977e-1
0.002355	0.002355 .002355L
$2.355 \cdot 10^{-3}$	2.355E-3 2.355e-3 .002355f

# Дійсні типи даних

Наприклад, якщо оголошуємо та ініціюємо змінну типу `float`, то:

```
float f1 =1.2f, af;
```

При оголошенні даних типу `long double` використовуємо суфікс `L` або `l`:

```
long double ldb, ldb1=1.234L, ldb2 = -1.e-10;
```

- Символ `e` або `E` відповідає  $10$ .

# Константи та літерали

## Константа

– це значення, що подаються в незмінюваному вигляді.

*Наприклад,*

- число **21** є цілою константою;
- число **3.14f** є константою дійсного типу **float**;
- символ **'%'** – це константа символьного типу **char**.

Такі незмінювані значення будемо називати **абсолютними константами** чи **літералами**.

# Константи та літерали

---

## Поіменовані константи

– константи, яким у коді надане ім'я або макро.

- Використовуються тоді, коли незмінювані значення – розміри статичних масивів, значення відомих констант або їх кратні величини, тощо.
- При цьому використання літералів потребує повторного втручання в код програми, що може призвести до появи помилок, які не відслідковуються при компіляції.

# Синтаксис оголошення констант

Поіменовані константи оголошуються (як і змінні)

- або за межами будь-якої функції;
- або в тілі функції (блоці).

Якщо константа оголошується, то їй слід одразу надати відповідне значення

`const` тип\_константи ім'я\_константи = значення;

або не ініціалізовувати, але оголосити з модифікатором `static`

`static const` тип\_константи ім'я\_константи;

`const int` size = 10, tru = 3;

`static const int` size1 = 11, nol; *//тут значення nol по замовчуванню рівне 0*

# Синтаксис оголошення констант

- Поіменовану константу цілого типу можна також оголосити, використовуючи перелічуваний тип:

```
enum {rozmir1 = 10, rozmir2 = 15};
```

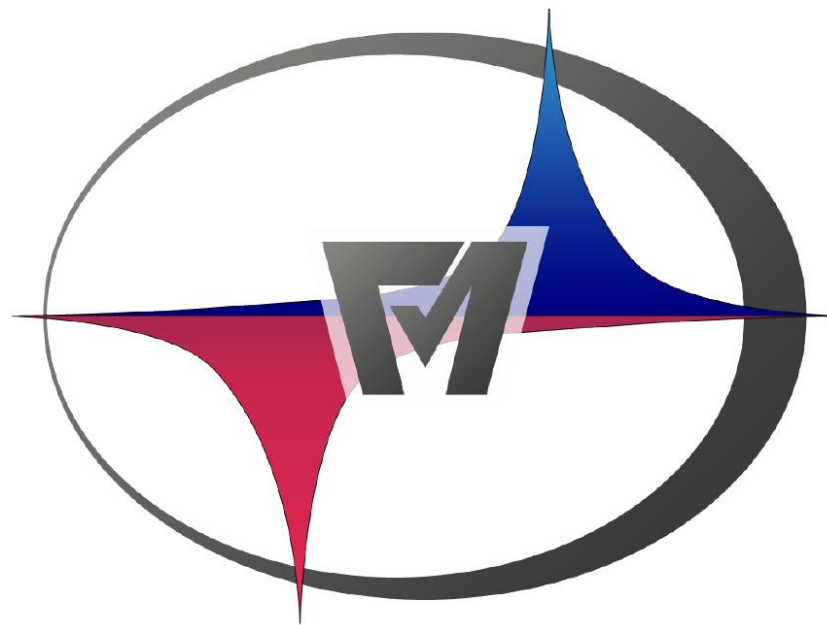
- На мові C задання констант можна здійснювати за допомогою директиви препроцесора `define`:

```
#define tak 1
```

```
#define ni 0
```

## Зауваження

*Якщо конструкція вимагає дане конкретного типу, то компілятор може видати застереження при виконанні чи навіть помилку виконання, тому константи краще задавати зі вказанням типу.*



Кафедра прикладної математики

<http://amath.lp.edu.ua>