Базові поняття мови С

Лекція №5

Дисципліна «Програмування»

Константи

Константа – це значення, яке не може бути змінено.

Синтаксис мови С визначає п’ять типів констант:

- символи;
- константи перелічуваного типу;
- дійсні числа;
- цілі числа;
- нульовий вказівник («**null**-вказівник»).

Стандарт ввів додатковий тип констант «символ розширеної форми представлення», але зараз його підтримує незначна кількість компіляторів.

Символи (символьні константи)

Для зображення окремих знаків, що мають індивідуальні внутрішні коди, використовуються **символьні константи**.

Кожна символьна константа – це лексема, яка складається з зображення символу та апострофів, які її обмежують. Наприклад: **'A'**, **'a'**, **'B'**, **'8'**, **'0'**, **'+'**, **';'**.

В ЕОМ використовуються також коди, які не мають графічного представлення на екрані дисплея, клавіатури або принтері.

Для їх зображення використовуються комбінації з декількох символів – **керувальні послідовності**.

Керуючі послідовності

'\n' – новий рядок;
'\t' – горизонтальна табуляція;
'\r' – повернення каретки (курсору) до початку рядка;
'\\' – зворотна коса риска ****;
'\'' – апостроф (символ «одинарні лапки»);
'\"' – лапки (символ «подвійні лапки»);
'\0' – нульовий символ;
'\a' – сигнал-дзвоник;
'\b' – повернення на одну позицію (на один символ);
'\f' – переведення (прогін) сторінки;
'\v' – вертикальна табуляція;
'\?' – знак питання.

Ескейп-послідовності

Керувальні послідовності є окремим випадком ескейп-послідовностей (**ESC-sequence**), до яких також належать лексеми виду **'\ddd'** або **'\xhh',** або **'\Xhh'**.

'\ddd' – вісімкове представлення будь-якої символьної константи, де буква **d** – це вісімкова цифра (від **0** до **7**). Наприклад, **'\017'** або **'\233'**.

'\xhh' або **'\Xhh'** – шістнадцяткове представлення будь-якої символьної константи, де **h** – це шістнадцяткова цифра (від **0** до **F**). Наприклад, **'\x0b'**, **'\x1A'** та інші.

Символьна константа (символ) має цілий тип, тобто символи можна використовувати в якості цілочислових операндів у виразах.

Цілі константи

Синтаксисом мови визначені **десяткові**, **шістнадцяткові** та **вісімкові** цілі константи. Основа системи числення визначається префіксом в запису константи. Для десяткових констант префікс не використовується.

Послідовність цифр, що починається з 0 та не містить десяткових цифр старше 7, сприймається як **вісімкова константа**: 016 – вісімкове представлення числа 14_{10}.

Послідовність шістнадцяткових цифр (0, 1, ..., 9, A, B, C, D, E, F), перед якою записані символи 0x або 0X, вважається **шістнадцятковою константою**:

0x16 – шістнадцяткове представлення числа 22_{10};

0XFF – шістнадцяткове представлення числа 255_{10}.

Кожна конкретна реалізація мови С вводить свої обмеження на граничні значення констант.

Дійсні константи

Для подання дійсних чисел використовуються **константи** у формі **з рухомою крапкою**. Кожна дійсна константа складається з наступних частин:

- ціла частина (десяткова ціла константа);
- десяткова крапка;
- дробова частина (десяткова ціла константа);
- ознака показника «**e**» або «**E**»;
- показник десяткової степені (десяткова ціла константа, можливо, зі знаком).

Під час запису констант з рухомою крапкою можуть опускатися ціла або дробова частина (але не одночасно); десяткова крапка або символ експоненти з показником степені (але не одночасно).

Приклади: 44. 3.14159 44e0 .314159E1 0.0

Граничні значення цілих констант

Діапазони значень констант			Тип даних
десяткові	вісімкові	шістнадцяткові	
от 0 до 32767	от 00 до 077777	от 0x0000 до 0x7FFF	int
–	от 0100000 до 0177777	от 0x8000 до 0xFFFF	unsigned int
от 32768 до 2147483647	от 020000 до 017777777777	от 0x10000 до 0x7FFFFFFF	long
от 2147483648 до 4294967295	от 020000000000 до 037777777777	от 0x80000000 до 0xFFFFFFFF	unsigned long
> 4294967295	> 037777777777	> 0xFFFFFFFF	помилка

Машинне представлення програми на мові С передбачає, що кожна константа, яка введена в програмі, займає в ЕОМ деяку комірку пам’яті. Розміри цієї комірки пам’яті та інтерпретація її вмісту визначаються типом відповідної константи.

Дійсні константи

Тип даних	Розмір, біт	Діапазон абсолютних величин
float	32	от 3.4E-38 до 3.4E+38
double	64	от 1.7E-308 до 1.7E+308
long double	80	от 3.4E-4932 до 1.1E+4932

Якщо програміста не влаштовує тип, який компілятор приписує константі, то тип можна явно вказати в запису константи за допомогою суфіксів:

F (або **f**) – **float** (для дійсних);
U (або **u**) – **unsigned** (для цілих);
L (або **l**) – **long** (для цілих і дійсних).

Наприклад:

3.14159F – константа типу **float** (виділяється 4 байта);
3.14L – константа типу **long double** (виділяється 10 байт).

За допомогою суфікса **U** (або **u**) можна подати цілу константу у вигляді беззнакового цілого.

Наприклад:

50000U – константа типу **unsigned int**.

Нульовий вказівник

Null-вказівник – це єдина не арифметична константа.

У конкретних реалізаціях **null**-вказівник може бути представлений або як **0**, або як **0L**, або як іменована константа **NULL**.

Значення константи **NULL** не повинно бути нулем, і має право не збігатися з кодом символу «**0**».

Константи перелічуваного типу

Цілочислові іменовані константи можна вводити за допомогою перелічення:

enum тип_перелічення { список_іменованих_констант };

де **enum** – службове слово, що вводить перелічення;
тип_перелічення – необов'язковий довільний ідентифікатор;
список_іменованих_констант – розділена комами послідовність ідентифікаторів або іменованих констант виду: ім'я_константи = значення константи

Приклади:

```
enum { ONE = 1, TWO, THREE, FOUR };
enum DAY { SUNDAY, MONDAY, TUESDAY, WEDNESDAY,
           THURSDAY, FRIDAY, SATURDAY };
enum BOOLEAN { NO, YES };
```

Рядки (рядкові константи)

Рядкова константа визначається як послідовність символів, яка **поміщається у подвійні лапки** (не в апострофи):

"Зразок рядка"

Серед символів рядка можуть бути ескейп-послідовності, тобто поєднання знаків, що відповідають символам, які не відображаються, або символам, що задаються їх внутрішніми кодами.

У цьому випадку, як і в представленнях окремих символьних констант, їх зображення починаються зі зворотної косої риски **'\'**:

"**\n** Текст **\n** розміститься **\n** в 3-х рядках дисплея"

Рядки (рядкові константи)

Кількість байтів, що виділяється в пам’яті ЕОМ для представлення значення рядка, рівно на **1** більше, ніж число символів у записі цієї рядкової константи:

"Цей рядок займає в пам\'яті ЕОМ 40 байт."

"Рядок у 17 байт."

При роботі з символьною інформацією потрібно пам’ятати, що довжина константи **'F'** дорівнює **1** байту, а довжина рядка **"F"** дорівнює **2** байтам.

Рядки (рядкові константи)

Якщо в послідовності символів (літер) константи зустрічається літера **'\'**, за якою до ознаки **'\n'** кінця рядка текстового файлу розміщені тільки пробіли, то ці пробіли разом з символом **'\'** і закінченням **'\n'** видаляються, і продовженням рядкової константи вважається наступний рядок тексту. Наприклад, наступний текст являє собою одну рядкову константу:

```
"Кафедра␣мультимедійних␣\
␣␣␣інформаційних␣технологій␣і␣систем."
```

У програмі ця константа буде еквівалентна такій константі:

```
"Кафедра␣мультимедійних␣␣␣␣інформаційних␣технологій␣і␣систем."
```

Початкові (ліві) пробіли в продовженні константи на новому рядку не видаляються, а вважаються такими, що входять до складу рядкової константи.

Рядки (рядкові константи)

Дві рядкові константи, між якими немає інших роздільників, крім узагальнених символів пробілів (пробіл, табуляція, кінець рядка та ін.); сприймаються як одна рядкова константа. Таким чином,

"Кафедра␣мультимедійних"␣"␣інформаційних␣технологій␣і␣систем."

сприймається як одна константа:

"Кафедра␣мультимедійних␣інформаційних␣технологій␣і␣систем."

Рядки (рядкові константи)

Таким самим правилам відповідають і рядкові константи, які розміщені на різних рядках. Як один рядок буде сприйнята послідовність

```
"Кафедра"
    "мультимедійних"
        "інформаційних"
            "технологій і систем."
```

Ці чотири рядкові константи еквівалентні одній:

```
"Кафедра мультимедійних інформаційних технологій і систем."
```

Змінна як об’єкт

Особливість змінної полягає в можливості пов’язувати з її ім’ям різні значення, сукупність яких визначається типом змінної.

При задаванні значення змінної у відповідну їй область пам’яті поміщається код цього значення.

Доступ до значення змінної забезпечує її ім’я, а **доступ до комірки пам’яті можливий тільки за її адресою.**

Визначення змінних

Кожна змінна перед її використанням у програмі повинна бути визначена, тобто для змінної повинна бути виділена пам’ять.

Розмір комірки пам’яті, що виділяється для змінної, і інтерпретація вмісту можуть відрізнятися залежно від типу, зазначеного у визначенні змінної.

Найпростіша форма визначення змінних:

тип список_імен_змінних;

де список_імен_змінних – це обрані програмістом ідентифікатори, які в списку розділяються комами;
тип – це тип змінної.

Типи даних

У версії **C89** визначено такі основні типи даних:

char	– символьний;
int	– цілочисловий;
float	– з рухомою крапкою;
double	– з рухомою крапкою подвійної точності;
void	– порожній.

До перелічених вище типів у версії **C99** додані наступні:

_Bool	– логічний або булевий (так/ні);
_Complex	– комплексний;
_Imaginary	– уявний.

Основні типи даних мови С

Тип		Систематичне ім'я	Інше ім'я
Цілий	беззнаковий	_Bool	bool
		unsigned char	
		unsigned short	
		unsigned int	unsigned
		unsigned long	
		unsigned long long	
	[без]знаковий	char	
	знаковий	signed char	
		signed short	short
		signed int	signed или int
		signed long	long
		signed long long	long long
З рухомою крапкою	дійсний	float	
		double	
		long double	
	комплексний	float _Complex	float complex
		float _Imaginary	float imaginary
		double _Complex	double complex
		double _Imaginary	double imaginary
		long double _Complex	long double complex
		long double _Imaginary	long double imaginary

Приклади визначень змінних

Приклади визначень цілочислових змінних:

```
char            symbol, cc;
unsigned char   code;
int             number, row;
unsigned long   long_number;
```

Стандартом мови введені такі дійсні типи:

float – дійсний тип одинарної точності;
double – дійсний тип подвійної точності;
long double – дійсний тип максимальної точності.

Приклади визначень дійсних змінних:

```
float     x, X, CC3, pot_8;
double    a, Stop, B4;
```

Граничні значення змінних

Тип даних	Розмір, біт	Діапазон значень
unsigned char	8	0 ... 255
char	8	-128 ... 127
enum	16	-32768 ... 32767
unsigned int	16	0 ... 65535
short int (short)	16	-32768 ... 32767
unsigned short	16	0 ... 65535
int	16	-32768 ... 32767
unsigned long	32	0 ... 4294967295
long	32	-2147483648 ... 2147483647
float	32	3.4E-38 ... 3.4E+38
double	64	1.7E-308 ... 1.7E+308
long double	80	3.4E-4932 ... 1.1E+4932

Ініціалізація змінних

Відповідно до синтаксису мови змінні автоматичної пам’яті після визначення за замовчуванням мають невизначені значення. Сподіватися на те, що вони дорівнюють, наприклад, **0**, не можна. Однак змінним можна присвоювати початкові значення, явно вказуючи їх у визначеннях:

тип ім'я_змінної = початкове_значення;

Приклади визначень з ініціалізацією:

```
float          pi = 3.1415, cc = 1.23;
unsigned int   year = 2019;
```

Іменовані константи

У мові С можуть бути визначені константи, що мають фіксовані назви (імена). У якості імен констант використовуються довільно обрані програмістом ідентифікатори, що не збігаються з ключовими словами та з іншими іменами об'єктів. Традиційно прийнято, що для позначень констант обирають ідентифікатори з ВЕЛИКИХ ЛІТЕР ЛАТИНСЬКОГО АЛФАВІТУ І СИМВОЛІВ ПІДКРЕСЛЕННЯ.

const тип ім'я_константи = значення_константи;

Приклади:

```
const double    E = 2.718282;
const long      M = 99999999;
const           F = 765;
```

Іменовані константи

Можливість вводити іменовані константи забезпечує препроцесорна директива

#define ім'я_константи значення_константи

```
#define     EULER2.718282

double      mix = EULER;
d = alfa * EULER;
```

Препроцесор замінить кожне позначення EULER на її значення і сформує такий текст:

```
double      mix = 2.718282;
d * = alfa * 2.718282;
```