



Java Starter

Машинна математика. Системи числення.

Java Starter

Після уроку обов'язково



Повторіть цей урок у форматі відео на
[ITVDN.com](http://itvdn.com)

Доступ можна отримати через керівництво
вашого навчального центру



Перевірте, як Ви засвоїли цей матеріал
на [TestProvider.com](http://testprovider.com)

Машинна математика. Системи числення

БіТ

Bit



Стани біта

State of bit

Не горить



✗

Неповністю горить



0

Горить на максимум



1

Один біт

Одним бітом можна представити 2 команди чи 2 числа



0

0 – З'єсти яблуко



1

1 – З'єсти грушку



Два біти

Двома бітами можна представити 4 команди або 4 числа



0 – з'їсти яблуко

0 0



1 – з'їсти грушку

0 1



2 – з'їсти сливку

1 0



3 – з'їсти ананас

1 1



Три біти

Трьома бітами можна представити 8 команд або 8 чисел



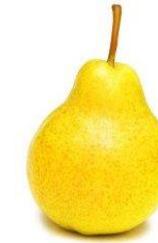
0 – З'єсти яблуко

0 0 0



1 – З'єсти грушку

0 0 1



...

...

...



7 – З'єсти полуницю

1 1 1



Чотири біти

Чотири бітами можна представити 16 команд або 16 чисел



0 0 0 0



0 0 0 1

...



1 1 1 1

0 – З'сти яблуко



1 – З'сти грушку



...

...

15 – З'сти банан



П'ять бітів

П'ятьма бітами можна представити 32 команди чи 32 числа



0 0 0 0 0

0 – з'єсти яблуко



0 0 0 0 1

1 – з'єсти грушку



• • •

• • •

• • •



1 1 1 1 1

31 – з'єсти персик



Шість бітів

Шістьма бітами можна представити 64 команди чи 64 числа



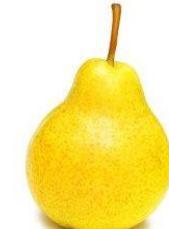
0 0 0 0 0 0

0 – 3'їсти яблуко



0 0 0 0 0 1

1 – 3'їсти грушку



• • •

• • •

• • •



1 1 1 1 1 1

63 – 3'їсти апельсин



Сім бітів

Сіома бітами можна представити 128 команд чи 128 чисел



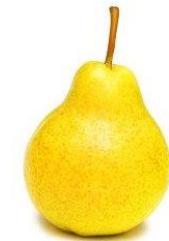
0 – З'єсти яблуко

0 0 0 0 0 0 0



1 – З'єсти грушку

0 0 0 0 0 0 1



• • •

• • •

• • •



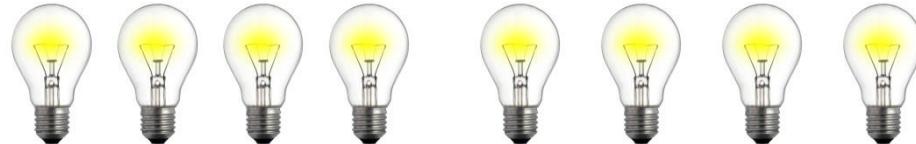
127 – З'єсти диню

1 1 1 1 1 1 1



Вісім бітів

Вісімома бітами можна представити 256 команд або 256 чисел



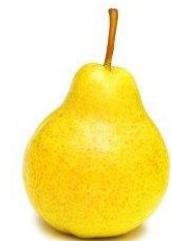
0 0 0 0 0 0 0 0

0 – 3'їсти яблуко



0 0 0 0 0 0 0 1

1 – 3'їсти грушку



...

...

...



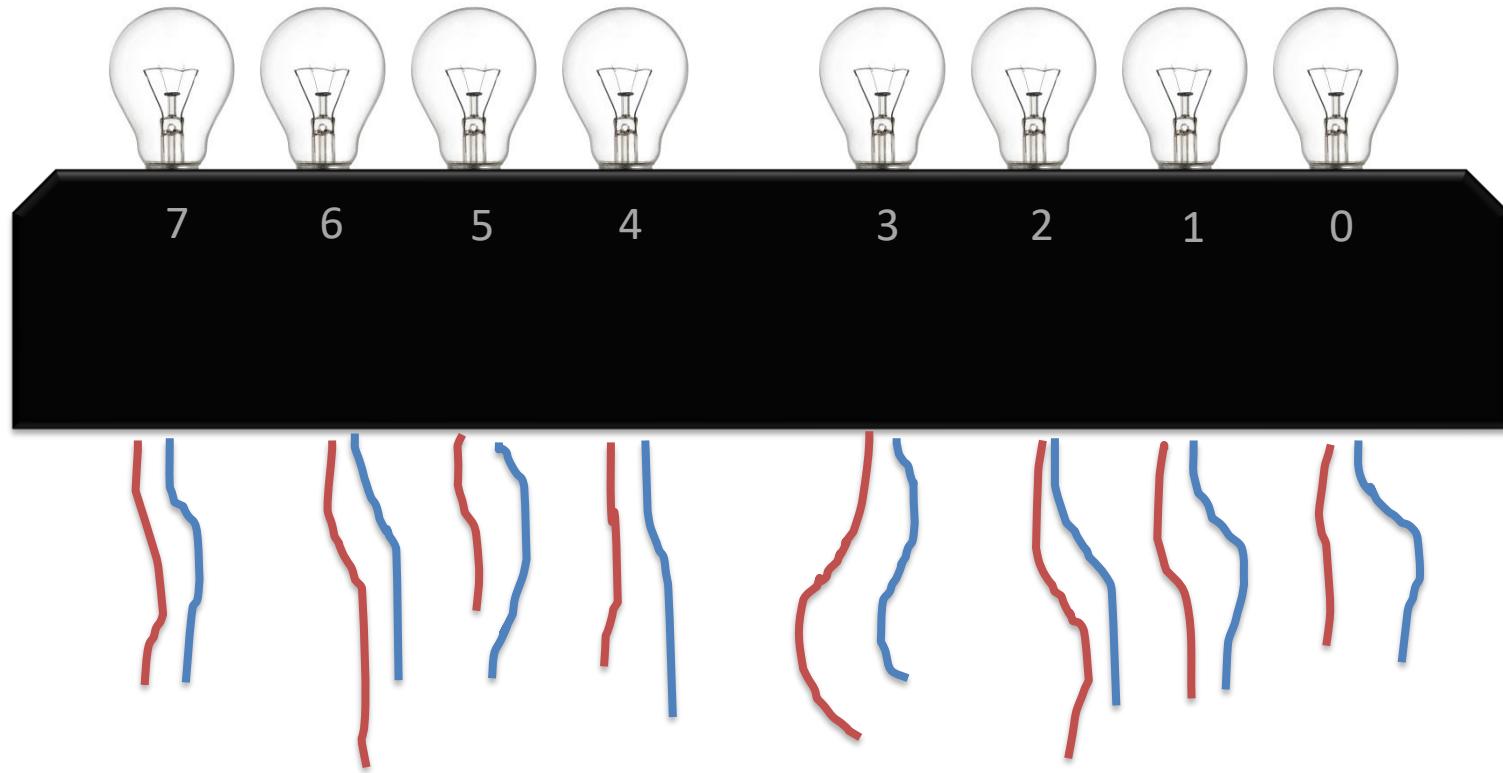
1 1 1 1 1 1 1 1

255 – 3'їсти кавун



Байт

1 байт = 8 бітів



Байт (byte) – одиниця зберігання та обробки цифрової інформації.

Одниці виміру кількості інформації

Units of data measurement

1 Кілобайт = 1024 Байти

1 Мегабайт = 1024 Кілобайти

1 Гігабайт = 1024 Мегабайти

1 Терабайт = 1024 Гігабайти

Одниці виміру кількості інформації

Units of data measurement



03П

Що всередині ОЗП?



ОЗП

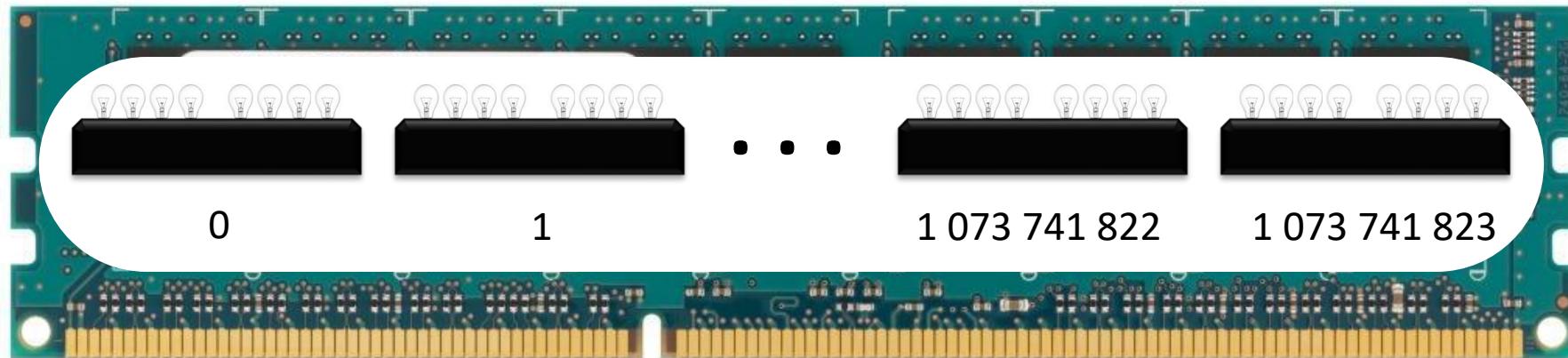
Якщо розмір ОЗП = 1 Гб, то в ньому є 1 073 741 824 Байти



$$1 \text{ GB} = 1024 \text{ MB} = 1024^3 \text{ B} = 2^{30} \text{ B} = 1\,073\,741\,824 \text{ B}$$

ОЗП

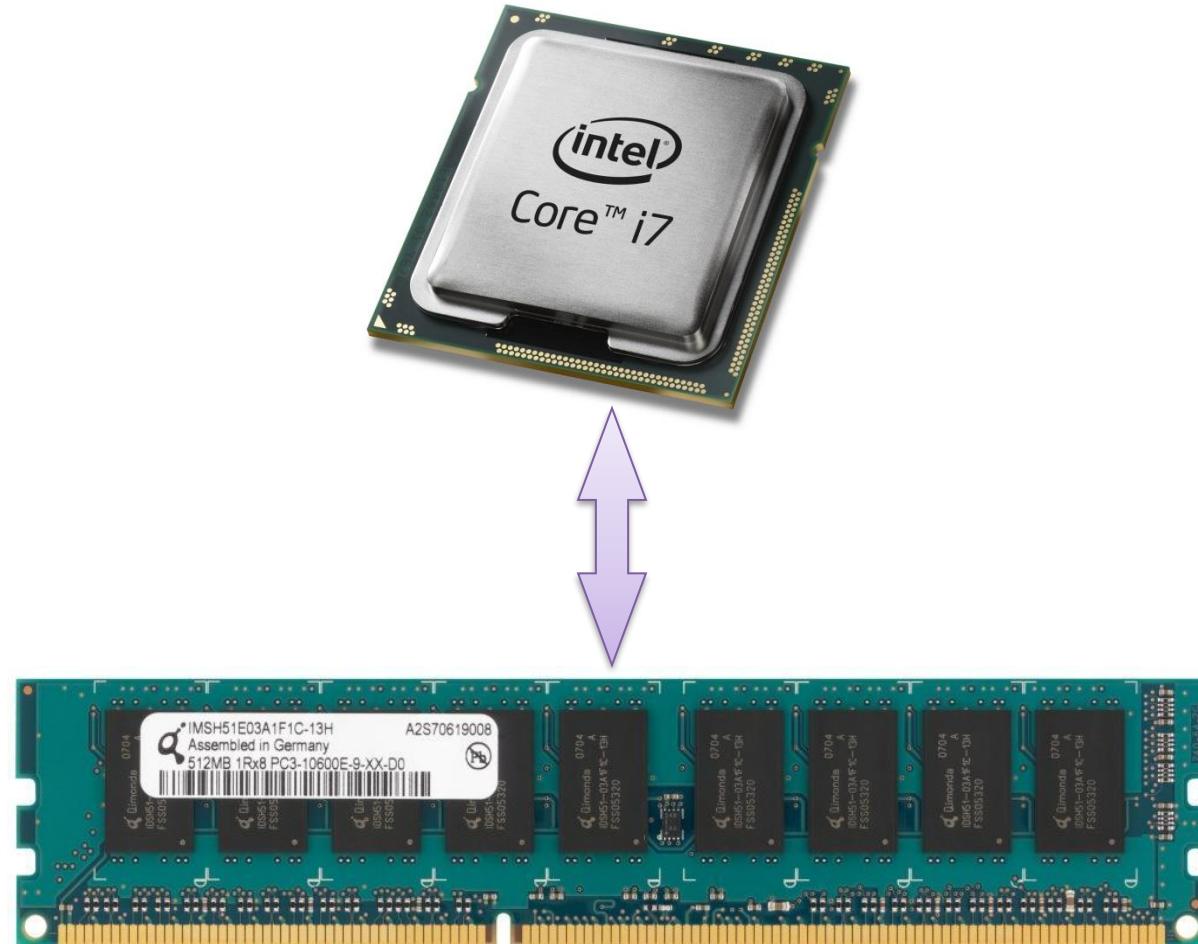
Процесор взаємодіє з пам'яттю



$$1 \text{ GB} = 1024 \text{ MB} = 1024^3 \text{ B} = 2^{30} \text{ B} = 1\ 073\ 741\ 824 \text{ B}$$

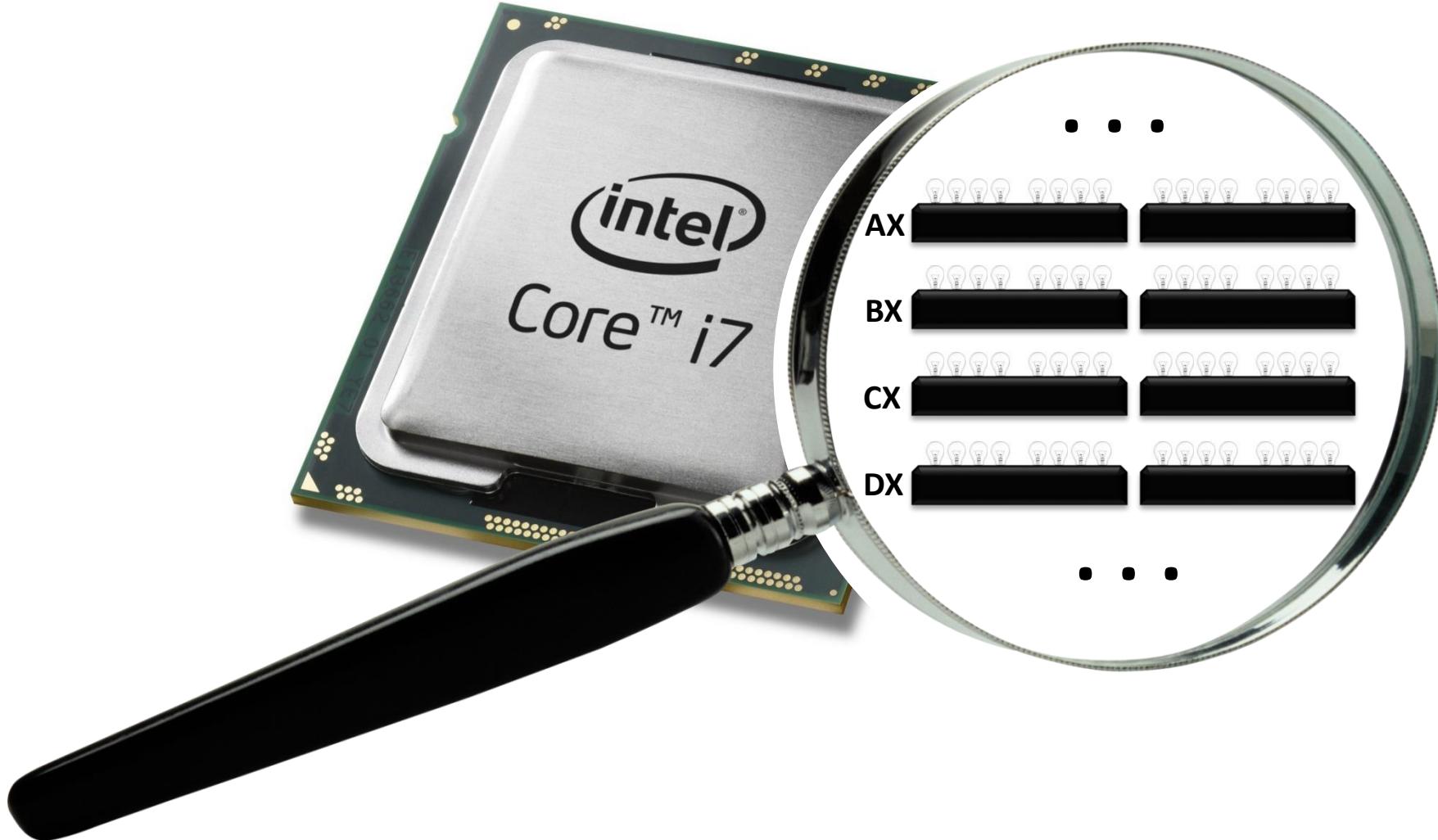
CPU взаємодіє з RAM

Процесор взаємодіє з пам'яттю



CPU

CPU має регістри, які подібні до осередків пам'яті



ОЗП

Варіанти зберігання інформації в ОЗП

1 байт = 8 бітів



2 байти = 16 бітів (Машинне слово)



4 байти = 32 біти (Подвійне машинне слово)



8 байтів = 64 біти (Учетверійне машинне слово)



Система числення

Символічний метод запису чисел

Система числення

Позиційна

значення кожного числового знака (цифри) у записі числа залежить від його позиції (розряду)

N2	N10	N16
0000 0000	0	00
0000 0001	1	01
0000 0010	2	02
0000 0011	3	03
0000 0100	4	04
0000 0101	5	05
0000 0110	6	06

Непозиційна

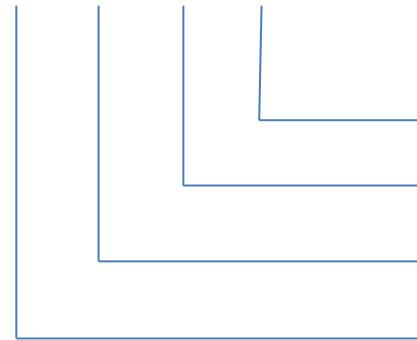
значення кожного символу не залежить від того місця, на якому він стоїть

1	2	3	4	5	6	7	8	9
а	в	г	д	е	с	ф	ү	ж
10	20	30	40	50	60	70	80	90
б	х	л	м	ү	է	0	π	ң
100	200	300	400	500	600	700	800	900
р	б	т	ә	ғ	ж	ұ	ұ	ә

Десяткова система числення

це позиційна система числення з основою 10

2886



одиниці
десятки
сотні
тисячі

Для запису числа використовуються цифри – 0123456789

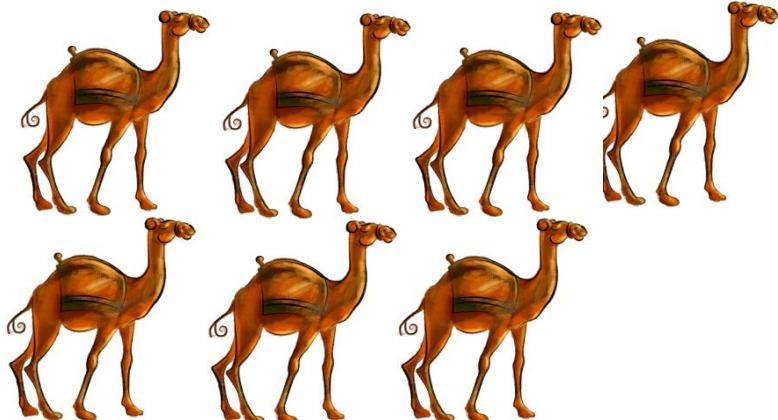
Основа системи числення – це число цифр у ній

П'ятирічна система числення

це позиційна система числення з основою 5



$$10_{10} \neq 10_5$$



\neq



Для запису числа використовуються цифри – 01234

$$10_5 = 5_{10}$$

Дванадцяткова система числення

це позиційна система числення з основою 12



10_{12} – Дюжина

100_{12} – Грос

Для запису числа використовуються цифри – 0123456789AB

$$10_{12} = 12_{10}$$

Двійкова система числення

це позиційна система числення з основою 2



Двійкова	Десяткова
0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8



Внутрішнє представлення будь-якої інформації у комп'ютері є двійковим.

Для запису числа використовуються цифри – 01

$$10_2 = 2_{10}$$

Шістнадцяткова система числення

це позиційна система числення з основою 16

Двійкова	Десяткова	Шістнадцяткова
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

Для запису числа використовуються цифри – 0123456789ABCDEF

$$10_{16} = 16_{10}$$

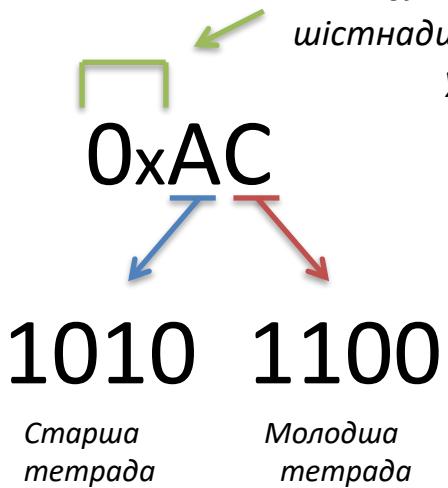
Зв'язок між шістнадцятковою та двійковою системами

Число у шістнадцятковому форматі можна представити у двійковому форматі та навпаки

Двійкова	Десяткова	Шістнадцяткова
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

Подайте кожен символ шістнадцяткового числа у вигляді тетради двійкових символів.

Префікс *0x* використовується для запису чисел у шістнадцятковому форматі у мові JAVA



Зв'язок між шістнадцятковою та двійковою системами

Число у двійковому форматі можна представити у шістнадцятковому форматі та навпаки

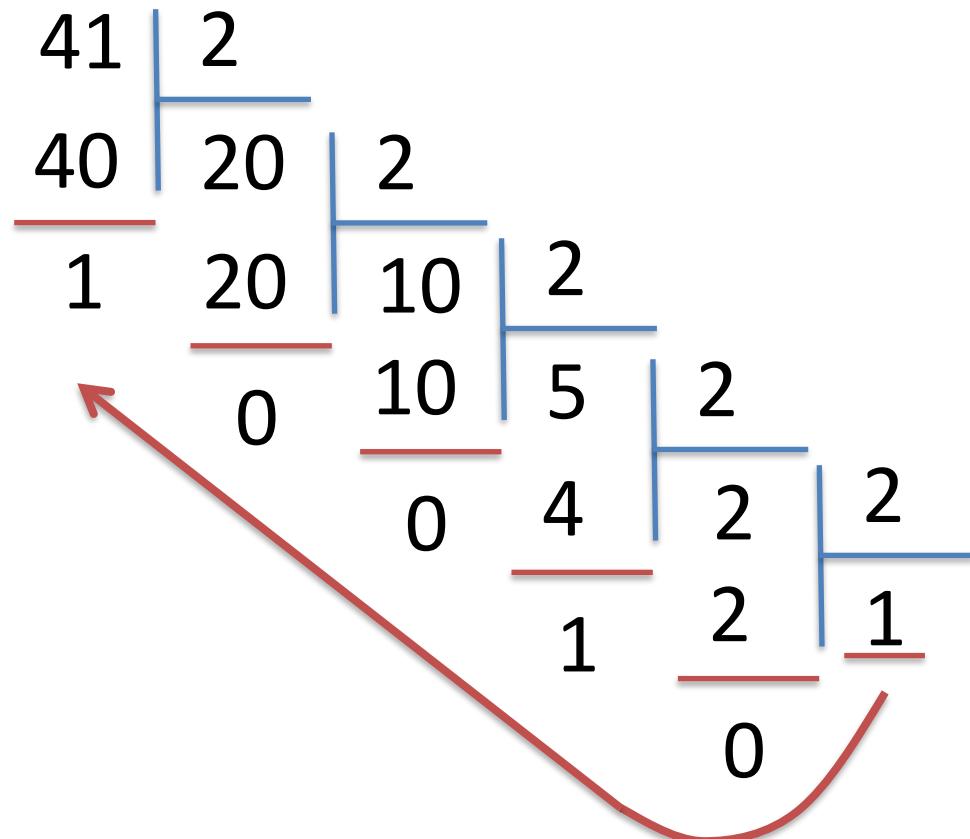
Двійкова	Десяткова	Шістнадцяткова
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

1110 1101
— —
0xED

Розділіть двійкове число на тетради та запишіть значення кожної тетради у шістнадцятковому поданні

Переведення десяткового числа у двійкове

Приклад



$$41_{10} = 101001_2$$

Щоб перетворити число, записане в десятковому форматі, на двійковий, необхідно:

- 1) послідовно ділiti задане число й отримані цiлi частини на 2 до тих пiр, поки цiла частина не стане меншою за 2;
- 2) отриманi залишки вiд подiлу, поданi цифрами з нового числення, записати як числа, починаючи з останньої цiлої частини (польська нотацiя).

Переведення двійкового числа в десяткове

Приклад

10 1001
Розряд 5 4 3 2 1 0

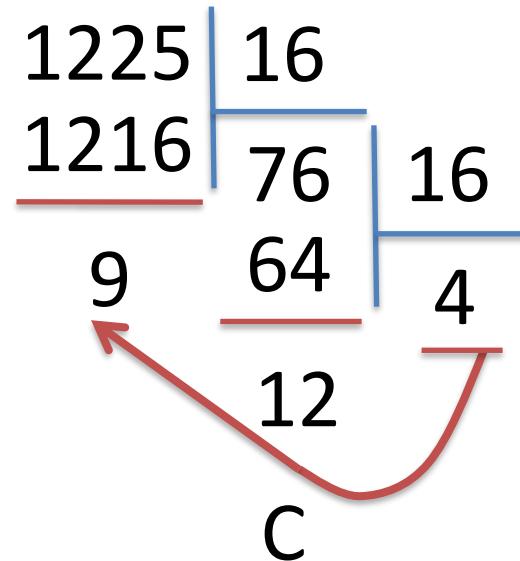
$$10\ 1001 = \\ \downarrow^5 \quad \downarrow^3 \quad \downarrow^0 \\ 2^5 + 2^3 + 2^0 = \\ 32 + 8 + 1 = 41_{10}$$

Щоб перетворити число, записане у двійковому форматі, на десятковий, необхідно:

- 1) замінити 1 у числі на 2, зведену до ступеня відповідного розряду цієї 1;
- 2) виконати додавання отриманих значень.

Переведення десяткового числа в шістнадцяткове

Приклад



$$1225_{10} = 4C9_{16}$$

Щоб перетворити число, записане в десятковому форматі, на шістнадцятковий, необхідно:

- 1) послідовно ділити задане число й отримані цілі частини на 16 до тих пір, поки ціла частина не стане меншою за 16;
- 2) отримані залишки від поділу, подані цифрами з нового числення, записати як числа, починаючи з останньої цілої частини.

Переведення шістнадцяткового числа в десяткове

Приклад

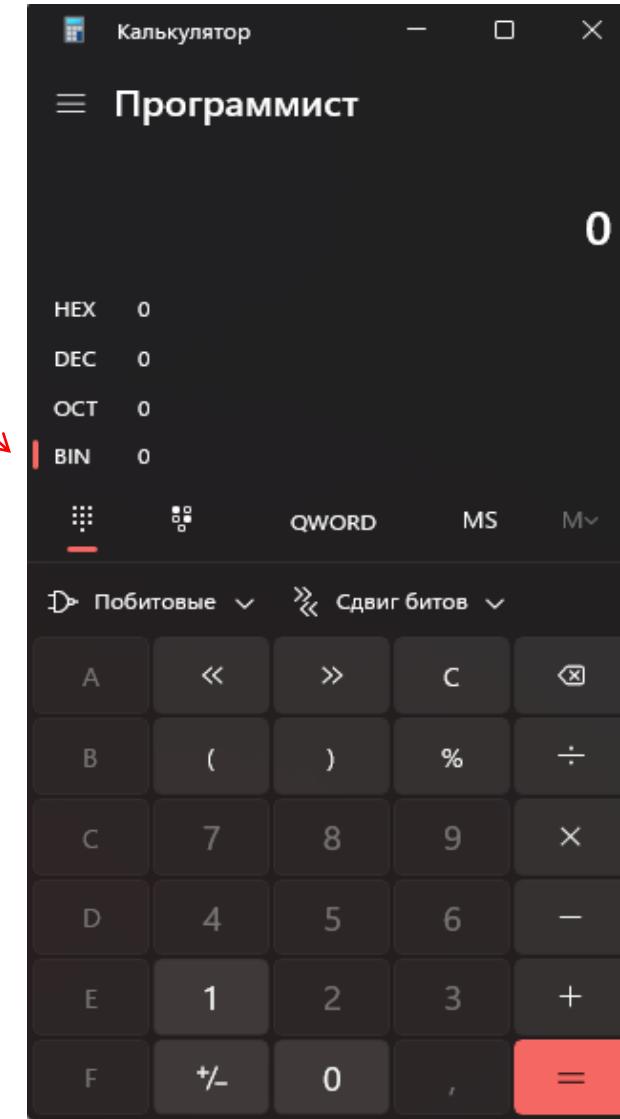
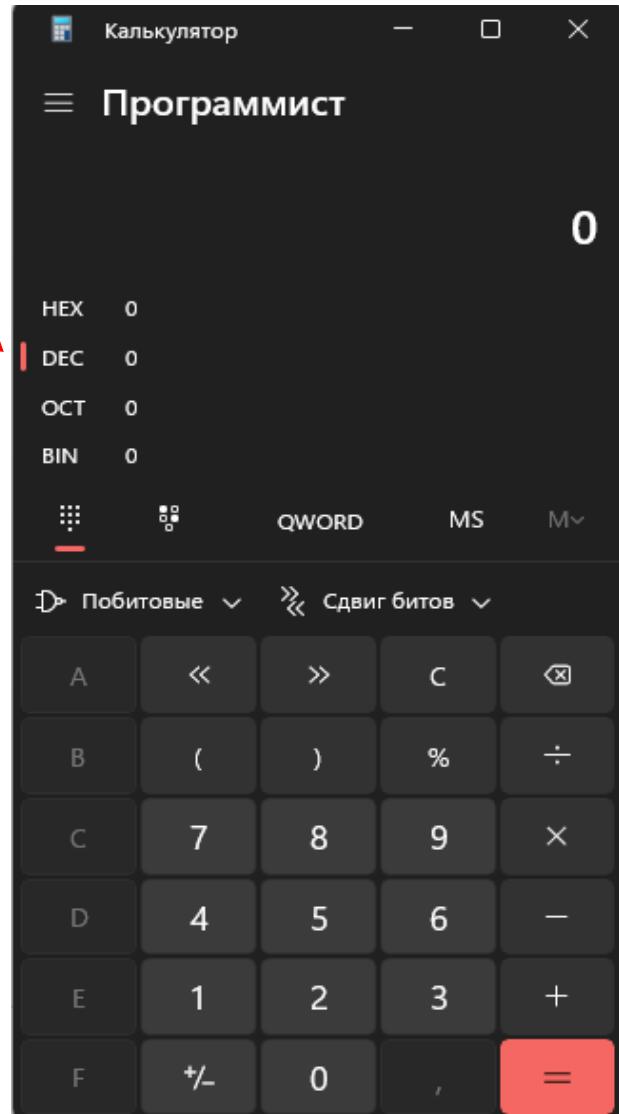
5A
Розряд 1 0

Щоб перетворити число, записане у шістнадцятковому форматі, на десятковий, необхідно:

$$A_{16} = 5 \cdot 16^1 + 10 \cdot 16^0 = 90_{10}$$

- 1) число помножити на 16 у ступені відповідно з розрядом;
- 2) виконати додавання отриманих значень.

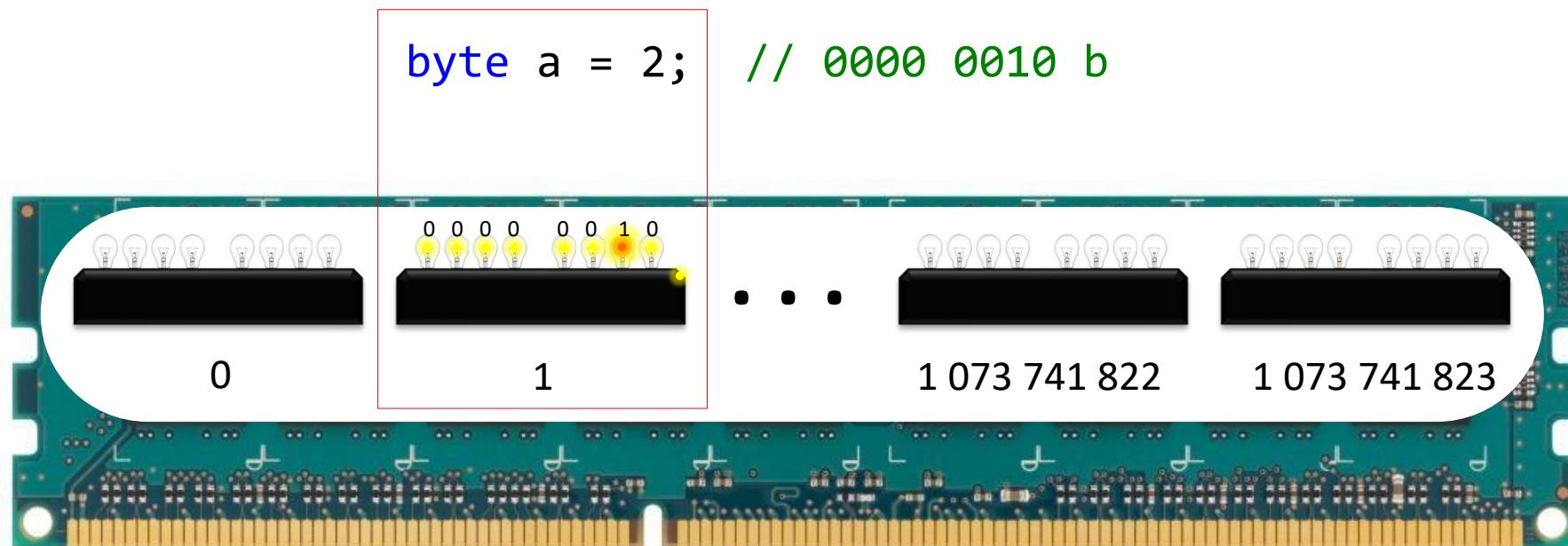
Використання калькулятора



Змінна

Variable

Змінна – це іменована частина пам'яті, яка зберігає у собі деяке значення, яке можна змінити.



Змінна

Створення змінної

Під час створення змінної необхідно вказати:

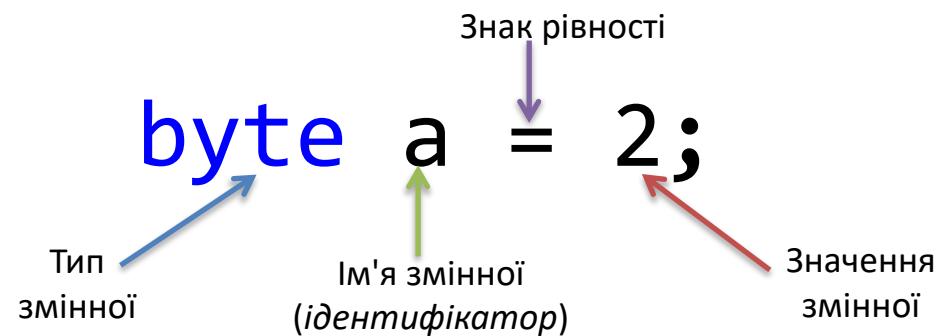
- ім'я змінної (*ідентифікатор*);
- тип змінної;
- початкове значення (*необов'язково*).

byte 1 byte from -128 to 127

short 2 bytes from -32,768 to 32,767

int 4 bytes -2,147,483,648 to 2,147,483,647

long 8 bytes from -9,223,372,036,854,775,808 to
9,223,372,036,854,775,807



Ініціалізація змінної – це перше присвоєння їй значення.

- **byte, short, int, long** зберігає цілі числа без десяткових знаків, наприклад, 123 або -123

Змінна

Створення змінної

byte (1 байт): від -128 до 127

short (2 байти): від -32,768 до 32,767

int (4 байти): від -2,147,483,648 до 2,147,483,647

long (8 байтів): від -9,223,372,036,854,775,808 до 9,223,372,036,854,775,807

float (4 байти): дробові числа. Достатньо для зберігання 6-7 десяткових цифр

double (8 байтів): дробові числа. Достатньо для зберігання 15 десяткових цифр

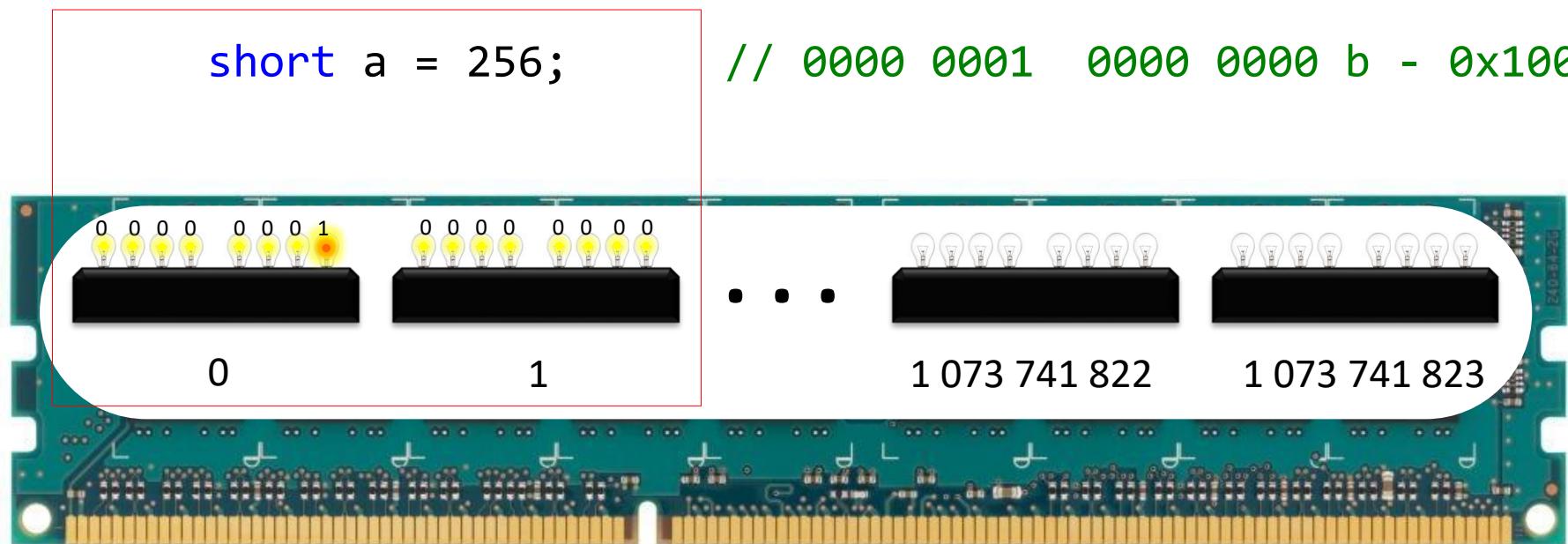
boolean (1 біт): значення true або false

char (2 біти): зберігає один символ/літер або значення ASCII

byte, short, int, long: зберігає цілі числа без десяткових знаків, наприклад, 123 або -123

Змінна Variable

Змінна – це іменована частина пам'яті, яка зберігає у собі деяке значення, яке можна змінити.



Типи даних ОЗП

Варіанти зберігання інформації у ОЗП

1 байт = 8 бітів byte boolean



2 байти = 16 бітів (Машинне слово) short char



4 байти = 32 біти (Подвійне машинне слово) int float

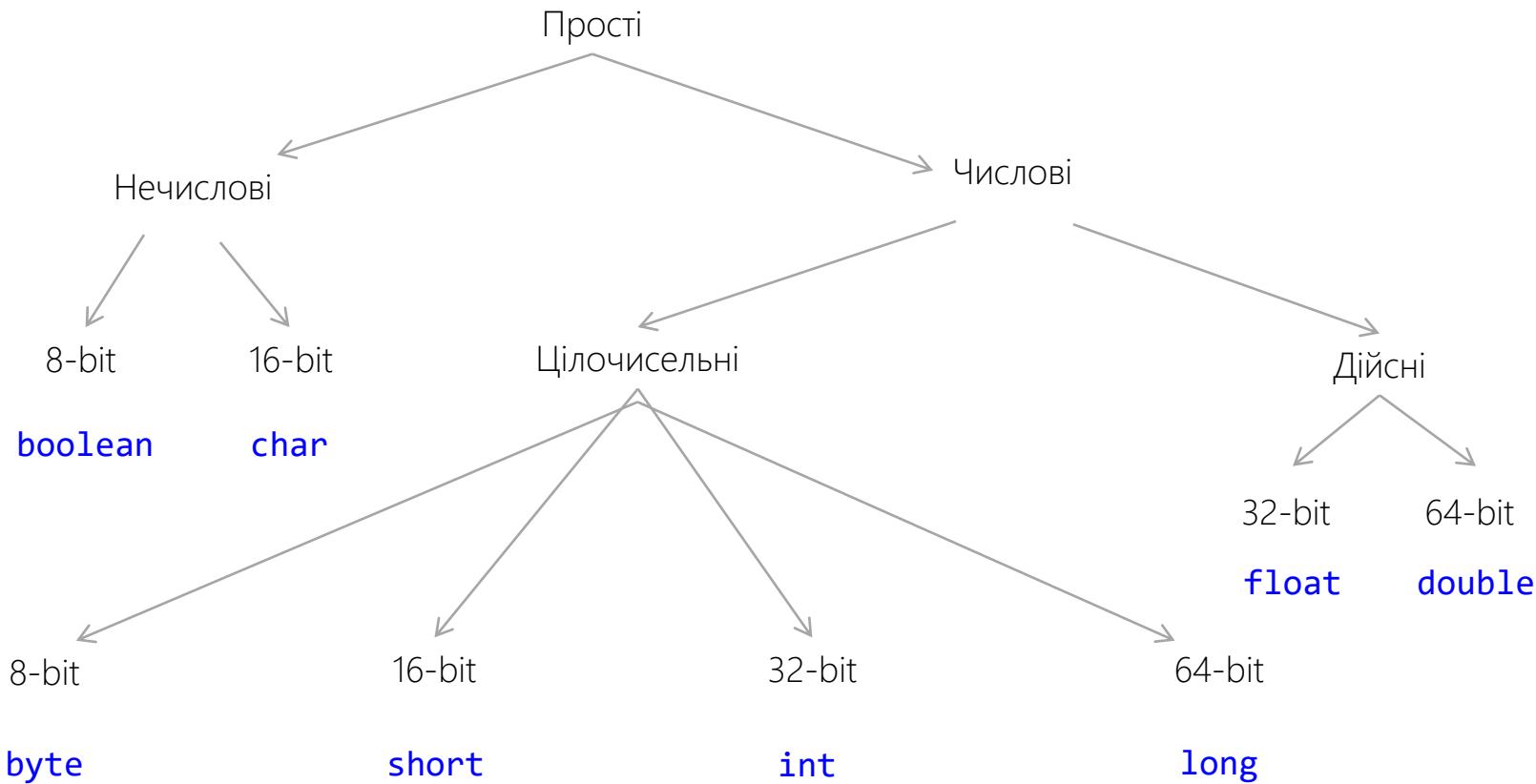


8 байтів = 64 біти (Учетверійне машинне слово) long double



Примітивні типи даних

Primitive Data Types



Q&A

Інформаційний відеосервіс для розробників програмного забезпечення

