

Основи програмування

Лектор

к.т.н. доцент кафедри програмних систем і технологій

КНУ ім. Тараса Шевченка

Ковалюк Тетяна Володимирівна

tkovalyuk@ukr.net

tetyana.kovalyuk@knu.ua

Лекція 1

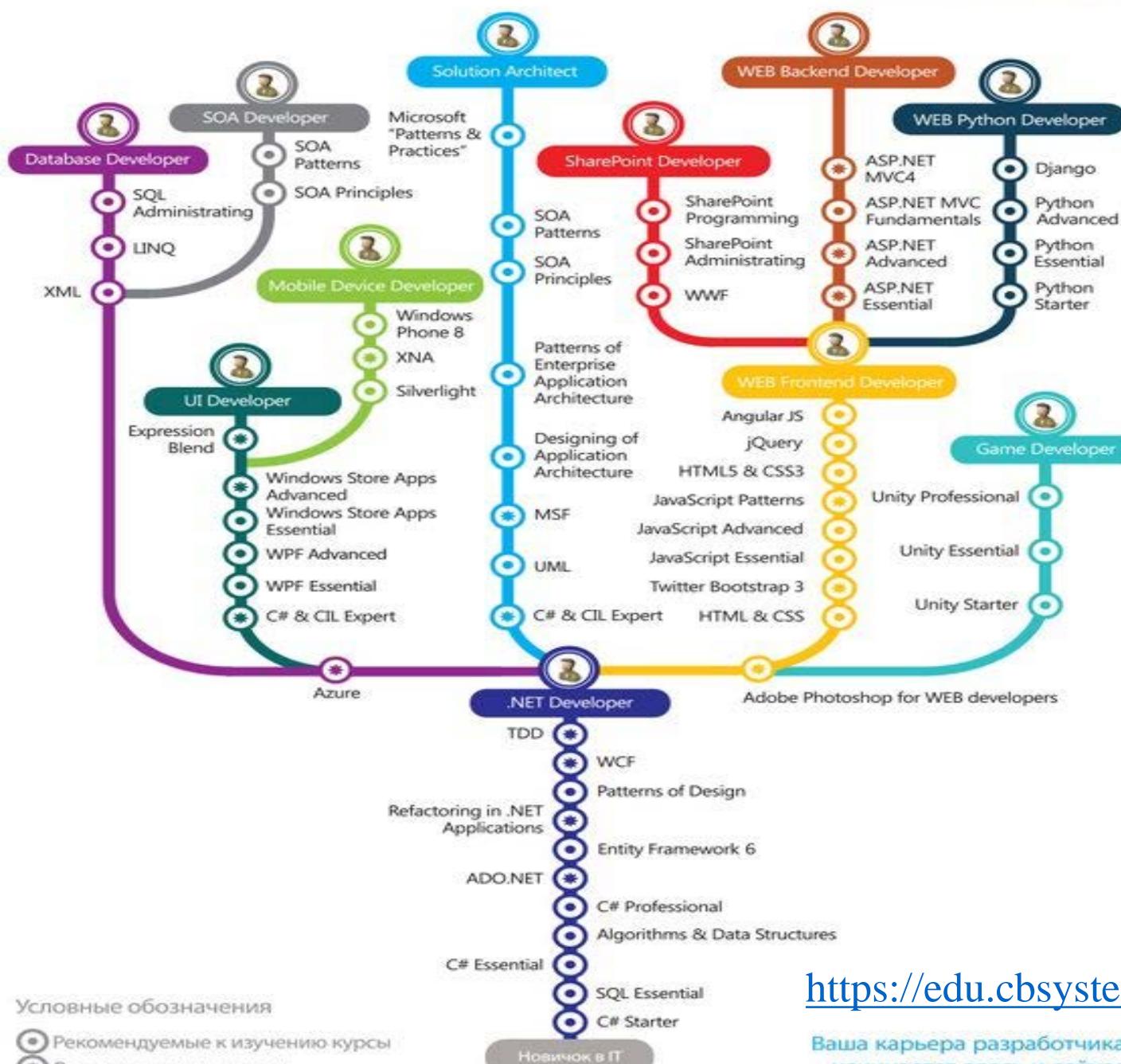
Вимоги до компетентностей студента з ІЗ.

Основні поняття програмування

Лектор к.т.н. доцент кафедри програмних систем і технологій

КНУ ім. Тараса Шевченка

Ковалюк Тетяна Володимирівна



C#/NET Full-stack Developer

Roadmap for Students


<https://edu.cbsystematics.com/ua/roadmap>

Ваша кар'єра разработчика
начинается здесь и сейчас

Професійні ІТ-стандарти для України



Базові знання фахівця з розробки програмного забезпечення



Загальні компетентності (soft skills) випускника з ІПЗ

1. Здатність до абстрактного мислення, аналізу та синтезу.
2. Здатність застосовувати знання у практичних ситуаціях.
3. Знання та розуміння предметної області та розуміння професійної діяльності.
4. Здатність спілкуватися державною мовою як усно, так і письмово.
5. Здатність спілкуватися іноземною мовою.
6. Здатність вчитися і оволодівати сучасними знаннями.
7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
8. Здатність генерувати нові ідеї (креативність).
9. Здатність працювати в команді.
10. Здатність бути критичним і самокритичним.
11. Здатність розробляти та управляти проектами.
12. Здатність приймати обґрунтовані рішення.
13. Здатність оцінювати та забезпечувати якість виконуваних робіт.
14. Визначеність і наполегливість щодо поставлених завдань і взятих обов'язків.
15. Здатність діяти на основі етичних міркувань



Лектор, автор
Ковалюк Тетяна Володимирівна
tkovalyuk@ukr.net

Структура курсу



Основна література

1. **Ковалюк Т.В.** Алгоритмізація та програмування. – Львів.: «Магнолія 2006», 2015. – 400 с.
2. **Дейтел Х. М., Дейтел П. Дж.** Как программировать на C++. — М.: ЗАО «Издательство БИНОМ», 2021, 1032 с.
3. **Кёниг, Э., Му Б.** Эффективное программирование на C++. — М. : Издательский дом «Вильямс», 2007. — 384 с.
4. **Седжвик Р.** Фундаментальные алгоритмы на C++. — К.: Издательство «ДиаСофт», 2018. – 1056 с.
5. **Шилдт Г.** Самоучитель C++. — СПб.: БХВ-Петербург, 2003. – 687 с.
6. **Вирт Н.** Алгоритмы + структуры данных = программы.— М.: Мир, 2018. — 406 с.

Частина 1. Процедурне програмування

Розділ 1 Основні поняття та означення в мові C/C++

Зміст

- 1.1. Поняття архітектури комп'ютера**
- 1.2. Принципи архітектури комп'ютерів фон Неймана**
- 1.3. Архітектура комп'ютеру**
- 1.4. Інформація в пам'яті комп'ютера**
 - 1.4.1. Позиційні системи числення**
- 1.5. Висновки**

Результати вивчення лекції 1

Знання:

- Принципи організації комп'ютера фон Неймана
- Функціональні елементи комп'ютера
- Подання чисел в позиційних системах числення
- Правила переведення чисел з одної системи числення в інші

Уміння

- Представляти дані в різних системах числення
- Визначати характеристики комп'ютера в залежності від вимог користувача

Поняття архітектури комп'ютера

Поняття архітектури обчислювальних систем є одним з основних для програмування

Під архітектурою комп'ютера розуміють логічну структуру в сукупності з фізичною структурною організацією



Поняття архітектури комп'ютера



В основу архітектури сучасних комп'ютерів покладено принципи, сформульовані в 1946 р. американським математиком та фізиком **Джоном фон Нейманом.**

Усі комп'ютери, побудовані згідно з цими принципами, відомі як **комп'ютери з фоннейманівською архітектурою.**

Принципи Фоннейманівської архітектури

**Використання
двійкової системи числення
для кодування інформації у
комп'ютері**

**Програмне керування
роботою комп'ютера**

**Принцип однорідності
пам'яті комп'ютера**

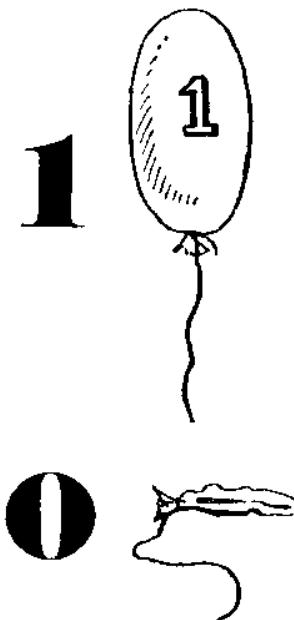
Адресація пам'яті

Принцип використання двійкової системи числення

Інформація (команди і дані) у комп'ютері кодуються у **двійковій системі числення.**

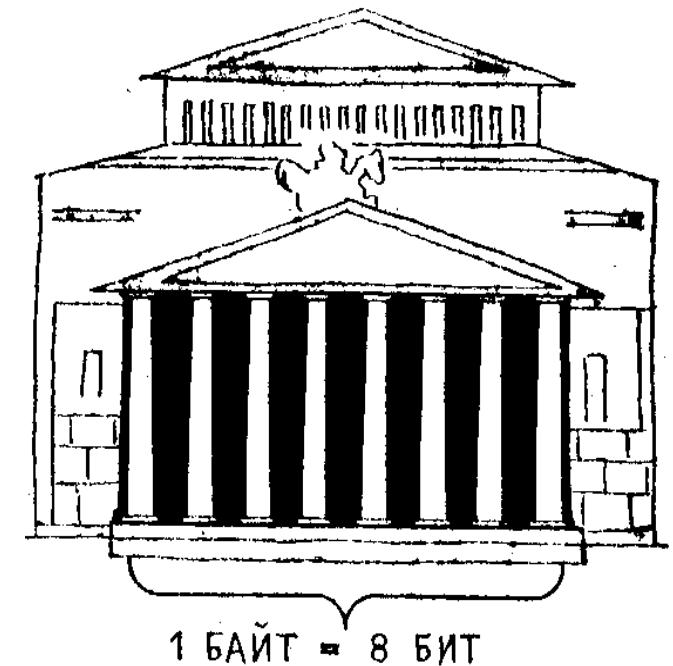
Система числення — це система позначення чисел.

У двійковій системі числення використовують дві цифри: **0 та 1.**



Принцип використання двійкової системи числення

- Одну двійкову цифру називають **бітом** (від англ. *binary digit* — двійкова цифра).
- Послідовність, що складається з восьми бітів, називають **байтом**.
- За допомогою одного байта можна закодувати $2^8 = 256$ різних повідомлень і зобразити значення цілих чисел із діапазону від **0 до 255**.



Принцип використання двійкової системи числення

Байт є одиницею обсягу пам'яті.

Для позначення тисяч, мільйонів та більше байтів використовують такі одиниці:

Назва одиниці	Позна чення	Значення в 10 с.с	Значення в 2 с.с	Значення в байтах
Байт	B	-	2^0	1
Кілобайт	kB	10^3	2^{10}	1024
Мегабайт	MB	10^6	2^{20}	1 048 576
Гігабайт	GB	10^9	2^{30}	1 073 741 824
Терабайт	TB	10^{12}	2^{40}	1 099 511 627 776
Петабайт	PB	10^{15}	2^{50}	1 125 899 906 842 624



Принцип однорідності пам'яті

Команди програми та дані, над якими виконуються дії, зберігаються в **одній і тій самій пам'яті** і зовні в пам'яті невиразні

Розпізнати команди і дані можна тільки **за способом використання**. Це дозволяє проводити над командами ті самі операції, що і над даними (числами)

Наслідок принципу однорідності – **команди однієї програми можуть бути отримані як результат виконання іншої програми**. Ця можливість лежить в основі трансляції

Принцип програмного керування

Усі обчислення, що виконує комп'ютер, зображені в пам'яті комп'ютера у вигляді **програм**.

Програма складається з послідовності **інструкцій – команд**.

Кожна команда наказує деяку **операцію** з набору операцій, реалізованих обчислювальною машиною.

Команди програми зберігаються в послідовних комірках пам'яті комп'ютера і **виконуються в порядку їх положення в програмі**.

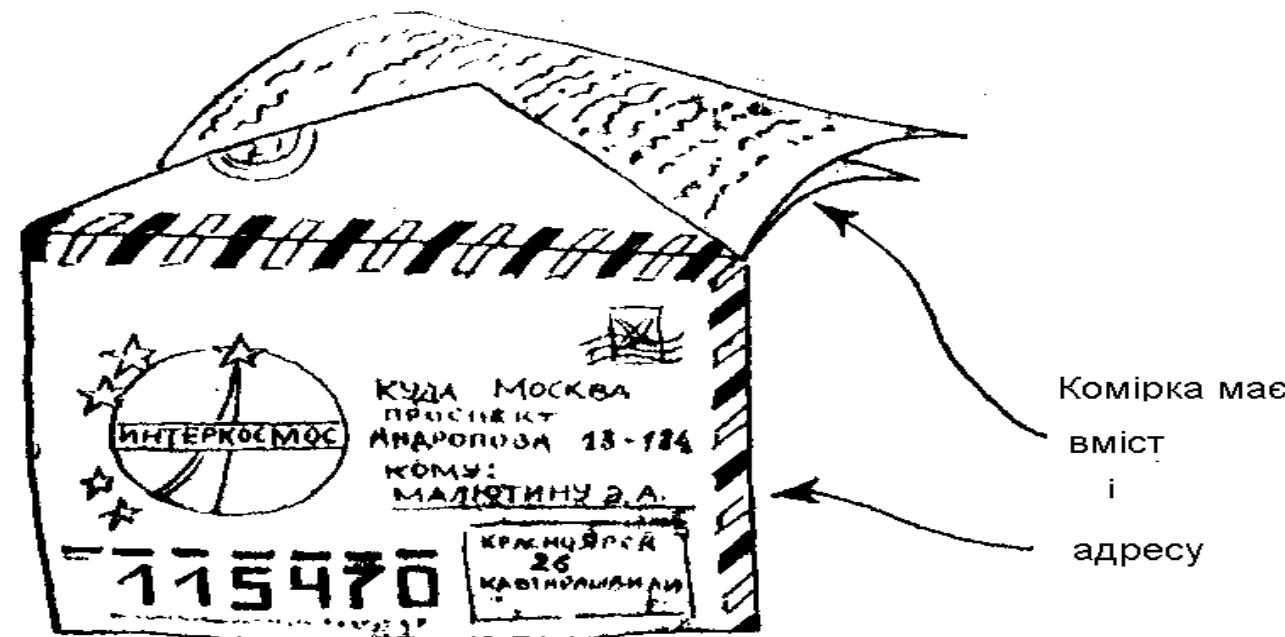
Принцип програмного керування

Програма керує комп'ютером



Принцип адресації пам'яті

Основна пам'ять – це послідовність комірок, кожна з яких має свій номер – адресу.
За адресою зберігаються певні дані (вміст адреси)



Принцип адресації пам'яті

Двійкові коди команд і даних поділяються на одиниці інформації, звані словами, і зберігаються в комірках пам'яті, а для доступу до них використовуються номери відповідних комірок **адреси**.

Мінімальний розмір комірки дорівнює восьми двійковим розрядам – **одному байту**.

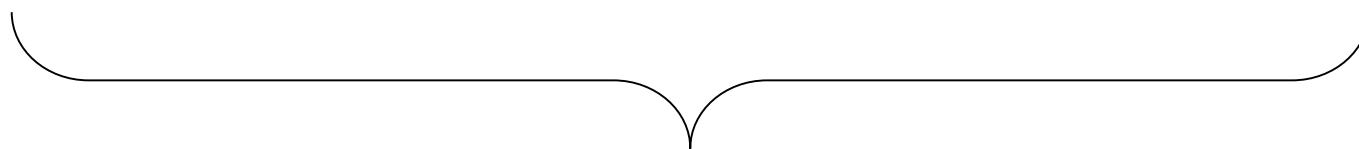
Memory dump

00000000	EF	BB	BF	3C	3F	78	6D	6C	20	76	65	72	73	69	6F	6E
00000010	3D	22	31	2E	30	22	20	65	6E	63	6F	64	69	6E	67	3D
00000020	22	55	54	46	2D	38	22	20	73	74	61	6E	64	61	6C	6F
00000030	6E	65	3D	22	79	65	73	22	3F	3E	0D	0A	3C	61	73	73
00000040	65	6D	62	6C	79	20	78	6D	6C	6E	73	3D	22	75	72	6E
00000050	3A	73	63	68	65	6D	61	73	2D	6D	69	63	72	6F	73	6F
00000060	66	74	2D	63	6F	6D	3A	61	73	6D	2E	76	31	22	20	6D
00000070	61	6E	69	66	65	73	74	56	65	72	73	69	6F	6E	3D	22
00000080	31	2E	30	22	3E	0D	0A	20	20	3C	61	73	73	65	6D	62
00000090	6C	79	49	64	65	6E	74	69	74	79	20	76	65	72	73	69
000000a0	6F	6E	3D	22	31	2E	30	2E	30	2E	30	22	20	6E	61	6D

Принцип адресації пам'яті

Адресою числа вважається **адреса першого байта** області пам'яті, відведеної ~~для~~ збереження числа

00000010 00000011 00000100 00000101

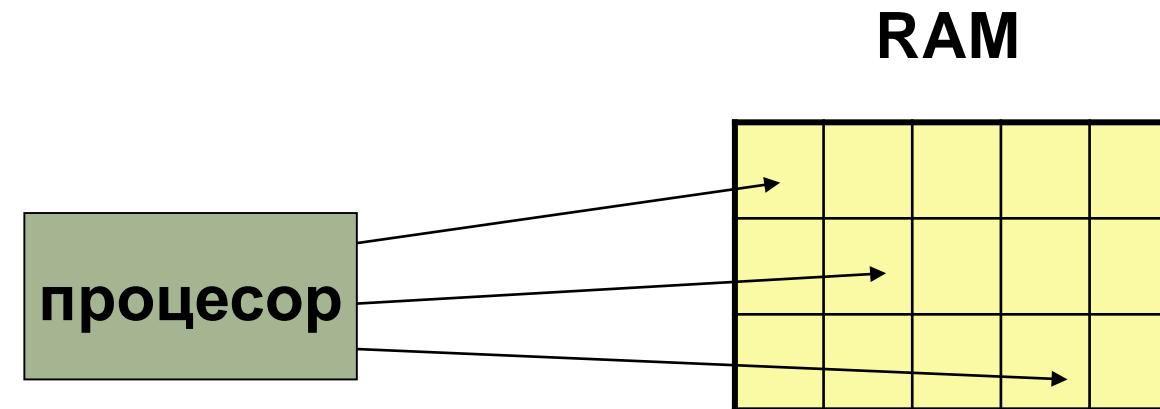


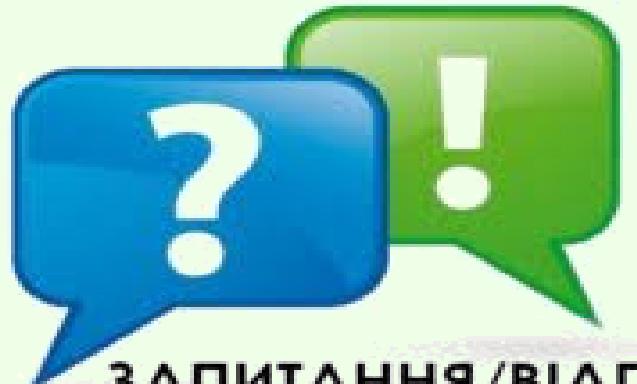
4 байти для одного числа

Принцип адресації пам'яті

Процесор у будь-який момент може отримати доступ до будь-якої комірки пам'яті.

Така пам'ять називається *пам'яттю з довільним доступом (RAM - random-access memory)*.



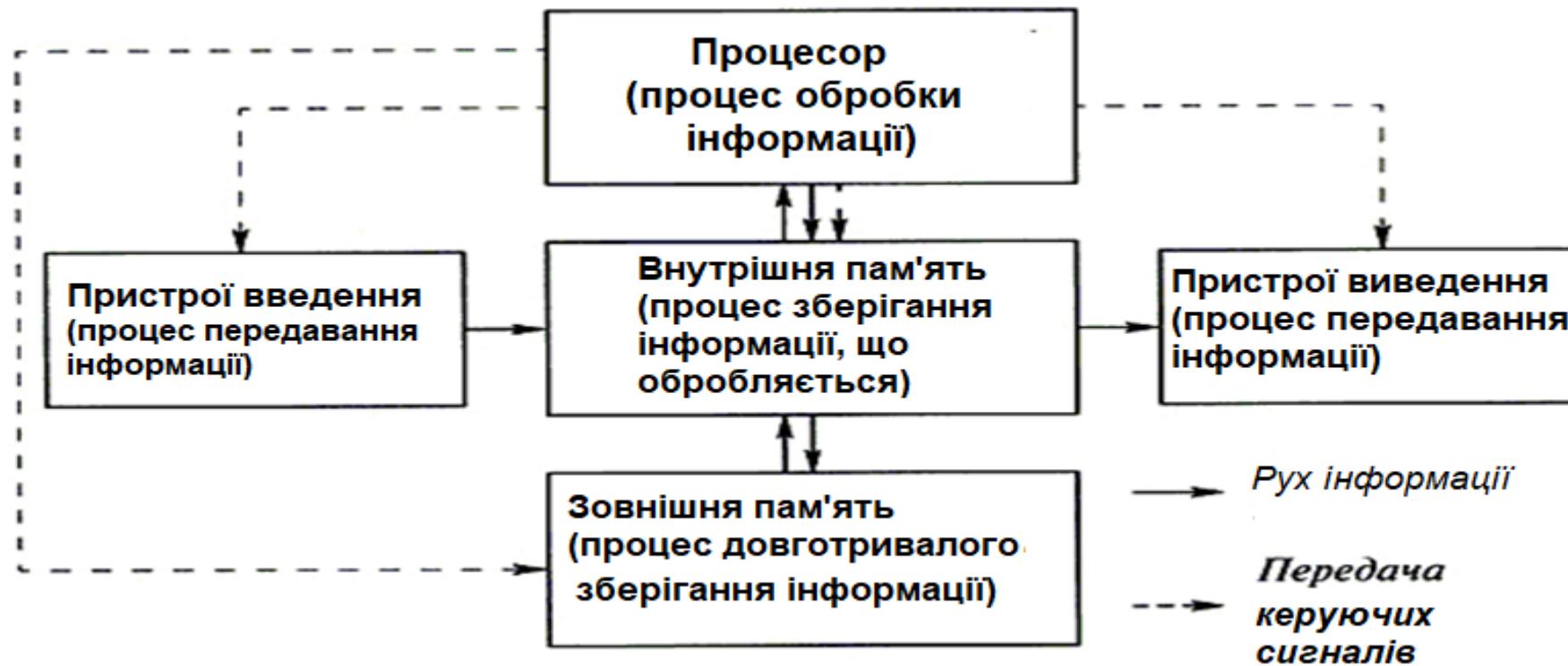


**Назвіть 4 принципи архітектури комп'ютера
Джона фон Неймана**

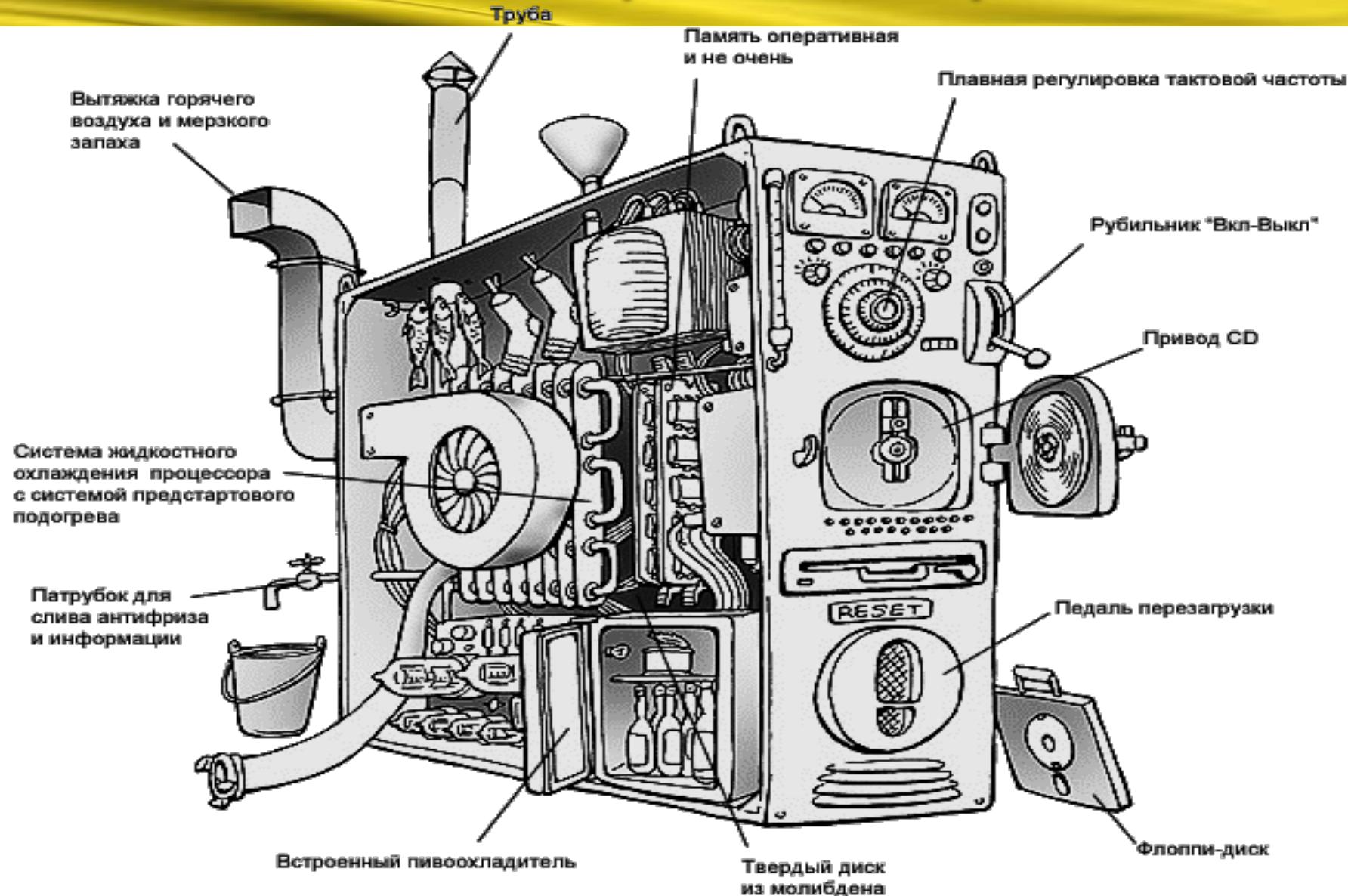
Архітектура комп'ютерів Джона фон Неймана

Типова фоннейманівська обчислювальна машина має такі складові:

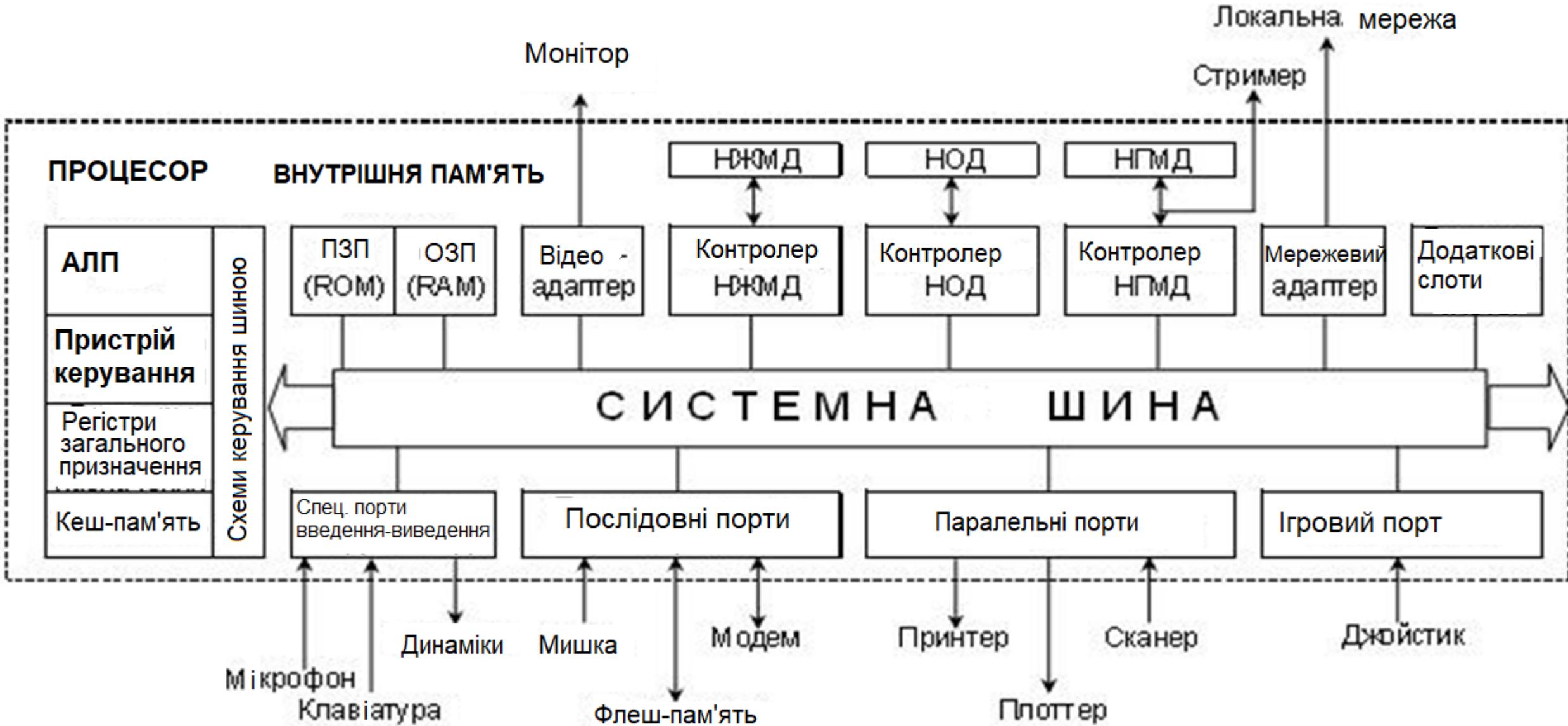
- ❖ **арифметико-логічний пристрій,**
- ❖ **пристрій керування,**
- ❖ **оперативна пам'ять**
- ❖ **пристрої введення-виведення.**



Фантазії на тему «Комп’ютер»



Функціональна схема комп'ютера



Архітектура комп'ютера



Арифметико-логічний пристрій (АЛП).



Функції АЛП

1. Виконання арифметичних, логічних та команд зсуву.
2. Забезпечення обробки вхідних даних та формування результату.

Конструктивно

АЛП — це група операційних пристрій для реалізації певних підмножин операцій

Пристрій керування (ПК)

Функції ПК

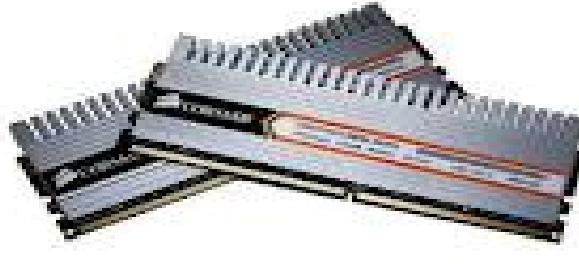
1. Координація роботи всіх пристрійв комп'ютера.
2. Формування сигналів для вибирання команд із пам'яті в порядку, що задається програмою.
3. Виконання команд програми.
4. Формування сигналів для синхронізації та координації дій зовнішніх і внутрішніх пристрійв комп'ютера.



Конструктивно

АЛП і ПК тісно взаємодіють між собою, і їх реалізують єдиним пристроєм, відомим як **центральний процесор**, чи просто **процесор**

Пам'ять комп'ютера



Функції пам'яті

1. Збереження інформації
2. Оперативний обмін інформацією з іншими компонентами комп'ютера.

Пам'ять комп'ютера

Види пам'яті

Внутрішня пам'ять складається з регистрів процесора, основної пам'яті та кеш-пам'яті.

Регістри процесора — це найбільш швидкодіючий, але найменший за обсягом різновид пам'яті комп'ютера. Регістри залежно від свого призначення можуть мати обсяг від 1 до 10 байт.

Основна пам'ять може включати пам'ять двох типів — **постійну й оперативну**.

Пам'ять комп'ютера

Види пам'яті

Постійна пам'ять (ROM) реалізується у вигляді постійного запам'ятовуючого пристрою.

Призначення ROM — підтримання процедур початкового завантаження операційної системи та обслуговування переривань.



Пам'ять комп'ютера

Види пам'яті

Оперативна пам'ять (RAM) використовується і для читання, і для запису інформації.

Під час роботи комп'ютера в *RAM* зберігаються програми та дані.

Оперативна пам'ять комп'ютера організована у вигляді множини байтів, або комірок, у яких зберігаються числові та символальні значення



Пам'ять комп'ютера

Види пам'яті

Кеш-пам'ять – це високошвидкісна пам'ять довільного доступу для тимчасового зберігання та прискорення обміну даними між процесором і RAM завдяки зменшенню кількості звертань до неї процесора.

Кеш-пам'ять первого рівня реалізується на **одному кристалі** з процесором і має об'єм в декілька десятків Кбайт,

Кеш-пам'ять другого рівня реалізується на **окремому кристалі**, але в межах процесора, з об'ємом в сто і більше Кбайт

Кеш-пам'ять третього рівня реалізується на **окремих швидкодійних мікросхемах** із розташуванням на материнській платі і має обсяг один і більше Мбайт.



Пам'ять комп'ютера

Види пам'яті

Зовнішня пам'ять призначена для довготривалого збереження великого обсягу даних.

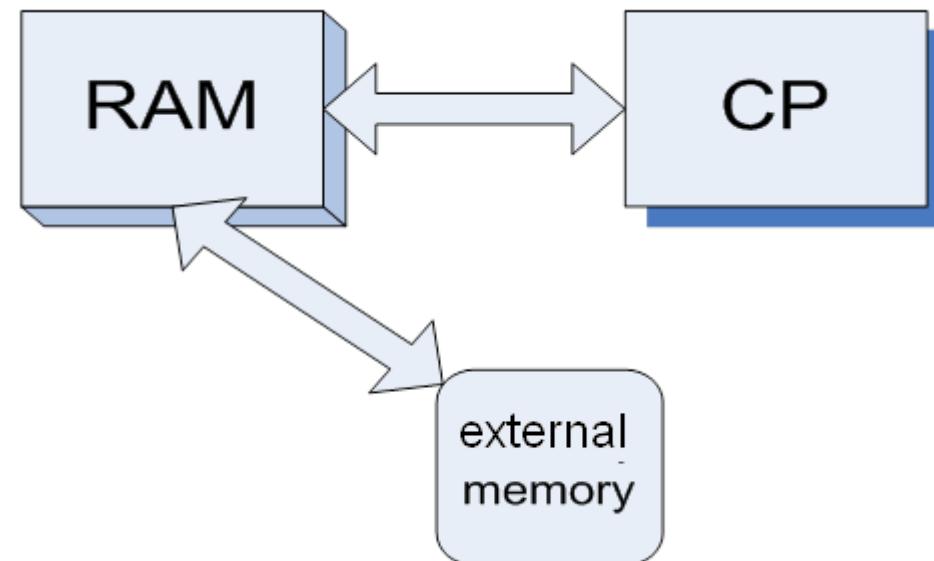
Склад зовнішньої пам'яті: накопичувачі на жорстких магнітних дисках (вінчестери), пристрой на касетній магнітній стрічці (стримери), накопичувачі на оптичних дисках CD, DVD-ROM/RW тощо.



Пам'ять комп'ютера

Взаємодія типів пам'яті

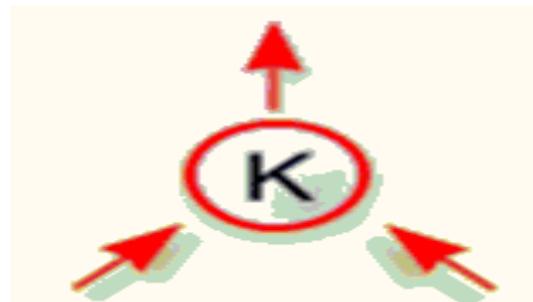
Інформація, яка зберігається у **зовнішній пам'яті (external memory)**, стане доступною для **процесора (CP)** тільки після того, як буде переписана в **основну пам'ять (RAM)**.



Пристрої введення-виведення

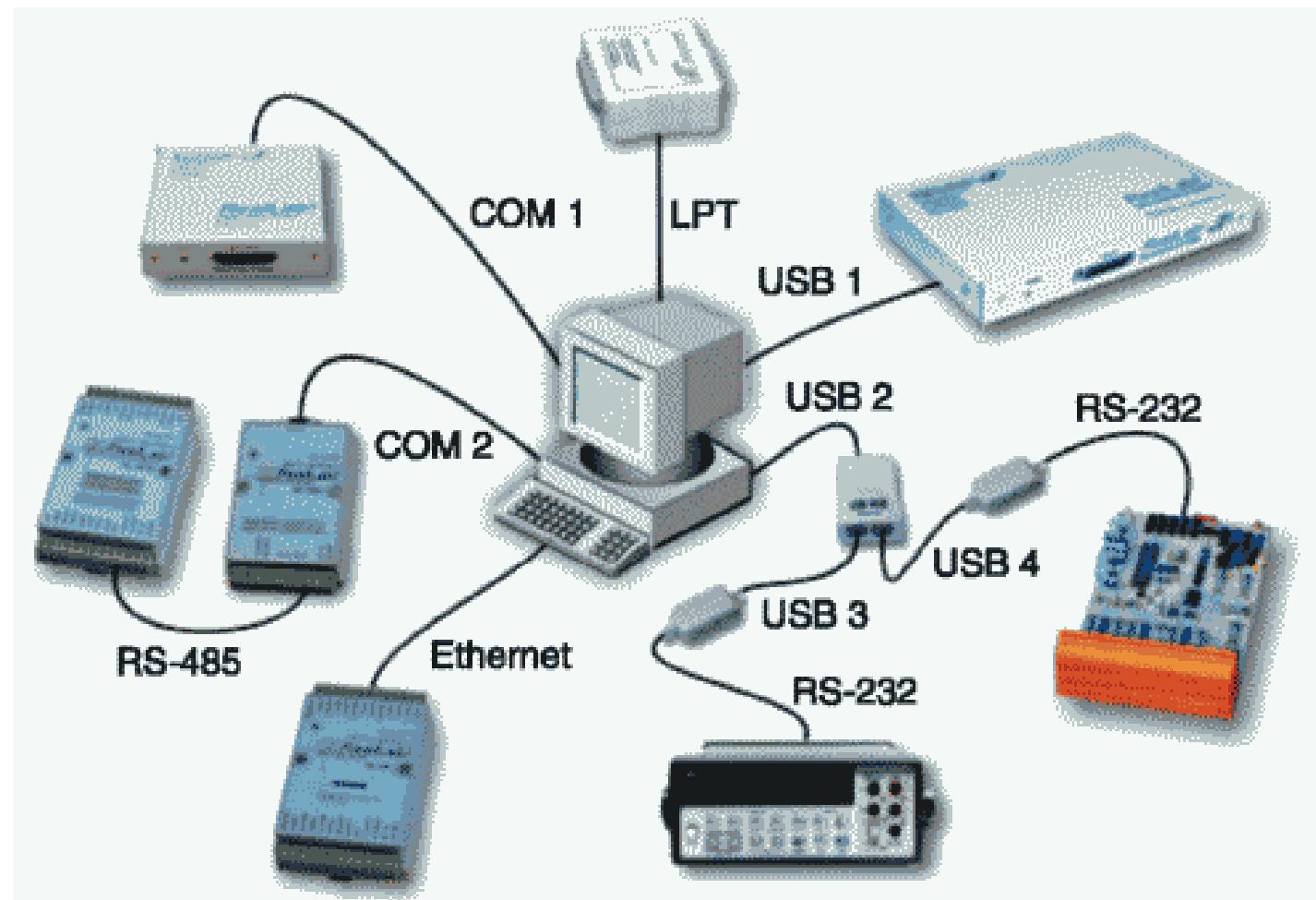
Забезпечують взаємодію комп'ютера з навколоишнім середовищем, користувачами, об'єктами керування та іншими комп'ютерами

Монітор, принтер, дисковод, флеш пам'ять, modem,



клавіатура, мишка, сканер

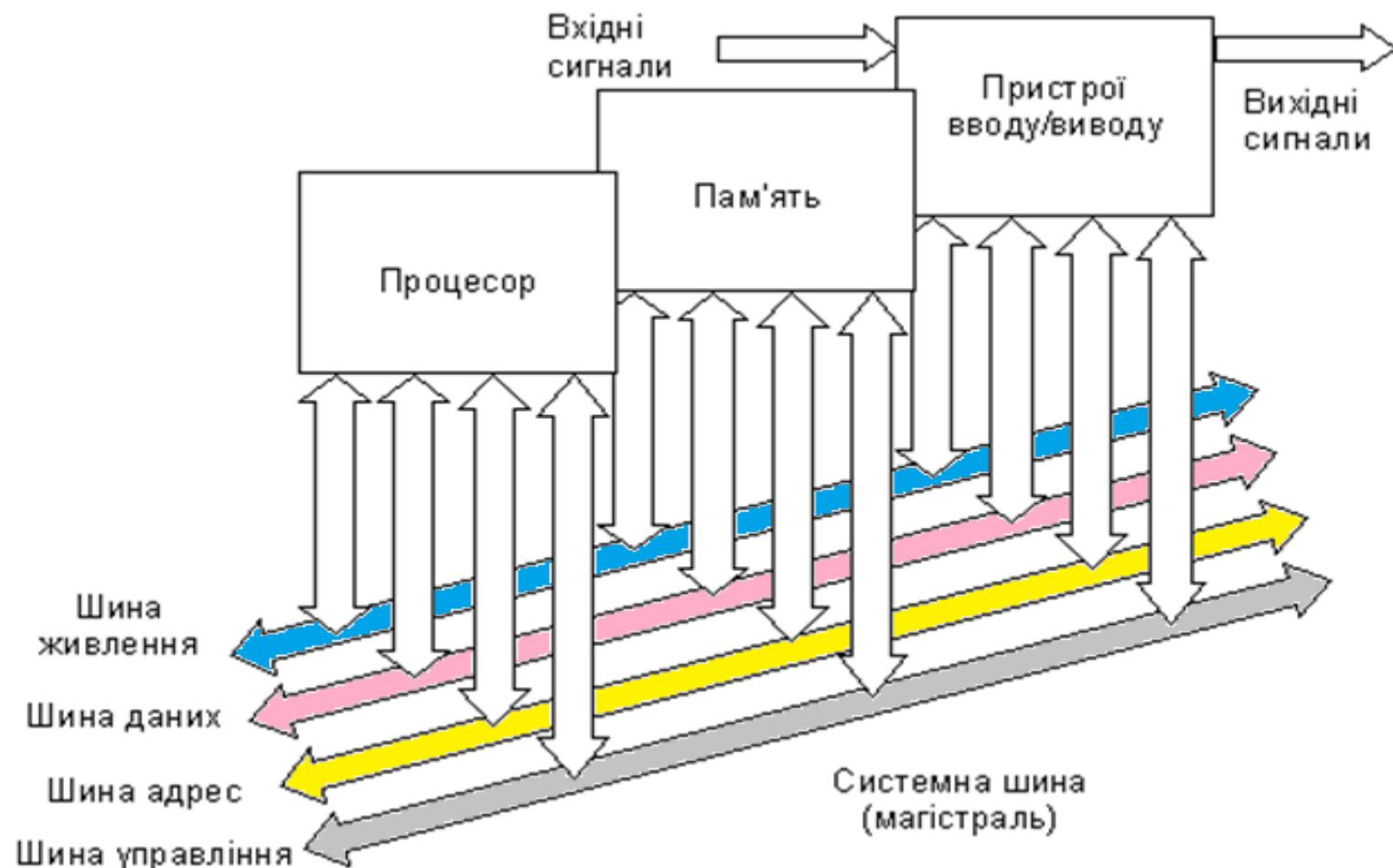
Пристрої введення-виведення



Пристрої введення-виведення

Група ліній зв'язку, якими передаються сигнали або коди саме і називається **шиною** (англ. bus).
Можна виокремити такі функціональні групи:

- шина адреси (Address Bus);
- шина даних (Data Bus);
- шина управління (Control Bus);
- шина живлення (Power Bus).





Назвіть 4 складові архітектури комп'ютерів Джона фон Неймана

Архітектура системи команд

Повний перелік команд, які здатен виконати певний комп'ютер, називають його **системою команд**.

Типова команда задає:

Операцію, яку слід виконати

Адреси даних (операндів), над якими має бути виконана операція;

Адресу пам'яті, де потрібно зберегти результат виконання операції.

КОП	A1	A2	A3
-----	----	----	----

Приклад машинної програми

Обчислити значення виразу

$$a = (b + c) \times d,$$

Значення змінних, a, b, c, d зберігається в адресах пам'яті

Ідентифікатор змінної	Призначення змінної	Адреса комірки оперативної пам'яті
a	Результат виразу	100
b	Операнд виразу	101
c	Операнд виразу	102
d	Операнд виразу	103
r	Проміжний результат	106

Приклад машинної програми

КОП	A1	A2	A3	Примітка
01	101	000	000	Увести значення змінної b до комірки з адресою 101
01	102	000	000	Увести значення змінної c до комірки з адресою 102
01	103	000	000	Увести значення змінної d до комірки з адресою 103
03	101	102	106	Додати значення змінних b і c , результат помістити в комірку з адресою 106, яка відповідає змінній r
05	106	103	100	Помножити значення змінних r і d , помістити результат у комірку з адресою 100 (у змінну a)
02	100	000	000	Вивести значення змінної a з комірки, що має адресу 100
00	000	000	000	Кінець обчислень (адреси операндів та результату не потрібні)

Інформація в пам'яті комп'ютера

Для зображення даних в пам'яті використовують системи числення :

- ✓ **двійкову** (використовує цифри 0 та 1),
- ✓ **шістнадцяткову** (використовує цифри 0.. 9 та символи A..F)
- ✓ **вісімкову** (використовує цифри 0..7)

Ці системи числення є **позиційними**, оскільки вага **кожної цифри** залежить від її **позиції** в записі числа.

Математично доведено, що числа з плаваючою точкою з основою 2 (двійкове подання) найбільш **стійкі до помилок округлення**, тому на практиці зустрічаються тільки основи 2

Позиційні системи числення

Значення Y числа, що записане у N -ковій системі числення, дорівнює значенню такого полінома

$$Y = \sum_{i=-m}^n x_i N^i$$

x_i — значення цифри в i -му розряді числа Y

i - i -ий розряд числа; $i = -1, -2, \dots, -m$ — розряди дробової частини числа;

$i = 0, 1, \dots, n$ — розряди цілої частини числа

N — система числення;

N^i - вага i -го розряду числа;

Позиційні системи числення

Приклади

$$512_{10} = 5 \times 10^2 + 1 \times 10^1 + 2 \times 10^0;$$

$$(512,346)_{10} = 5 \times 10^2 + 1 \times 10^1 + 2 \times 10^0 + 3 \times 10^{-1} + 4 \times 10^{-2} + 6 \times 10^{-3};$$

$$(10\ 011)_2 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 19_{10};$$

$$(10,011)_2 = 1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} = 2,375_{10}$$

$$(1BC)_{16} = 1 \times 16^2 + 11 \times 16 + 12 = 444_{10};$$

$$(1B,C)_{16} = 1 \times 16^1 + 11 \times 16^0 + 12 \times 16^{-1} = 27,75_{10};$$

$$(12,2)_3 = 1 \times 3^1 + 2 \times 3^0 + 2 \times 3^{-1} = 5,(6)_{10}$$

Зв'язок двійкової, вісімкової та шістнадцяткової систем

10 с.ч.	2 с.ч.	16 с.ч.	8 с.ч.
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11

10 с.ч.	2 с.ч.	16 с.ч.	8 с.ч.
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17
16	10000	10	20

	1	1	0	0	→	12
+				1	→	1
	1	1	0	1	→	13

Перенесення в старший розряд **1 1 1 1** **10 с ч**

1 1 1 1	10 с ч
	15
+	1
1 0 0 0 0	16

Календар на будь-який рік у 16-тковій с.ч

0x7DA

01

001 010 011 100 101 110 111
01 02 03
04 05 06 07 08 09 0a
0b 0c 0d 0e 0f 10 11
12 13 14 15 16 17 18
19 1a 1b 1c 1d 1e 1f

02

001 010 011 100 101 110 111
01 02 03 04 05 06 07
08 09 0a 0b 0c 0d 0e
0f 10 11 12 13 14 15
16 17 18 19 1a 1b 1c

03

001 010 011 100 101 110 111
01 02 03 04 05 06 07
08 09 0a 0b 0c 0d 0e
0f 10 11 12 13 14 15
16 17 18 19 1a 1b 1c
1d 1e 1f

04

001 010 011 100 101 110 111
01 02 03 04
05 06 07 08 09 0a 0b
0c 0d 0e 0f 10 11 12
13 14 15 16 17 18 19
1a 1b 1c 1d 1e

05

001 010 011 100 101 110 111
01 02
03 04 05 06 07 08 09
0a 0b 0c 0d 0e 0f 10
11 12 13 14 15 16 17
18 19 1a 1b 1c 1d 1e
1f

06

001 010 011 100 101 110 111
01 02 03 04 05 06
07 08 09 0a 0b 0c 0d
0e 0f 10 11 12 13 14
15 16 17 18 19 1a 1b
1c 1d 1e

07

001 010 011 100 101 110 111
01 02 03 04
05 06 07 08 09 0a 0b
0c 0d 0e 0f 10 11 12
13 14 15 16 17 18 19
1a 1b 1c 1d 1e 1f

08

001 010 011 100 101 110 111
01
02 03 04 05 06 07 08
09 0a 0b 0c 0d 0e 0f
10 11 12 13 14 15 16
17 18 19 1a 1b 1c 1d
1e 1f

09

001 010 011 100 101 110 111
01 02 03 04 05
06 07 08 09 0a 0b 0c
0d 0e 0f 10 11 12 13
14 15 16 17 18 19 1a
1b 1c 1d 1e

0A

001 010 011 100 101 110 111
01 02 03
04 05 06 07 08 09 0a
0b 0c 0d 0e 0f 10 11
12 13 14 15 16 17 18
19 1a 1b 1c 1d 1e 1f

0B

001 010 011 100 101 110 111
01 02 03 04 05 06 07
08 09 0a 0b 0c 0d 0e
0f 10 11 12 13 14 15
16 17 18 19 1a 1b 1c
1d 1e

0C

001 010 011 100 101 110 111
01 02 03 04 05
06 07 08 09 0a 0b 0c
0d 0e 0f 10 11 12 13
14 15 16 17 18 19 1a
1b 1c 1d 1e 1f

Переведення натуральних чисел з однієї системи числення до іншої

Перевести число P з M -кової системи числення до N -кової.

$$\begin{aligned}P &= x_k \times N^k + x_{k-1} \times N^{k-1} + \dots + x_1 \times N + x_0 \\&= ((x_k \times N + x_{k-1}) \times N + \dots + x_1) \times N + x_0\end{aligned}$$

Звідси видно, що значенням **наймолодшої цифри** x_0 є **остача від ділення** числа P на основу N
(усі операції здійснюються над M -ковими числами).

Переведення натуральних чисел з однієї системи числення до іншої

$$\begin{aligned}P &= x_k \times N^k + x_{k-1} \times N^{k-1} + \dots + x_1 \times N + x_0 \\&= ((x_k \times N + x_{k-1}) \times N + \dots + x_1) \times N + x_0\end{aligned}$$

Значенням другої справа цифри x_1 буде **остача від ділення частки**, яку отримано на попередній ітерації, на основу N .

Продовжуючи ці міркування, отримаємо циклічну процедуру, кожна ітерація якої полягатиме у **знаходженні частки та остатці від ділення деякого числа Q на основу системи числення N** .

Переведення натуральних чисел з однієї системи числення до іншої

Приклад

Перевести число 25_{10} з десяткової системи числення до двійкової.

$$25 : 2 = 12(1);$$

$$12 : 2 = 6(0);$$

$$6 : 2 = 3(0);$$

$$3 : 2 = 1(1).$$

Остання частка менша двох.

Остачі від ділення (у дужках)

Остання частка є **старшою (першою) цифрою** двійкового числа, до якого треба додати **остачі** у порядку, зворотному до порядку їх отримання.

Перша остача є молодшою (останньою) цифрою двійкового числа

Результатом є число 11001_2 .

Перевірка:

$$\begin{aligned}11001_2 &= 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\&= 16+8+1=25_{10}\end{aligned}$$

25_{10}	$2^4=16$	$25-16=9$ $2^3=8$	$9-8=1$ $2^2=4$ $4>1$	$2^1=2$ $2>1$	$2^0=1$
розряди	4	3	2	1	0
Цифри числа	1	1	0	0	1

Приклад

Перевести число 25_{10} з десяткової системи числення до двійкової.

$$42 : 2 = 21(0);$$

$$21 : 2 = 10(1);$$

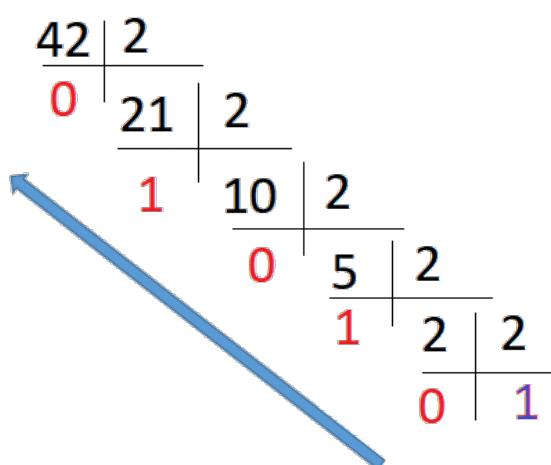
$$10 : 2 = 5(0);$$

$$5 : 2 = 2(1).$$

$$2 : 2 = 1(0)$$

Перевірка:

$$\begin{aligned} 101010_2 &= 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 \\ &+ 0 \times 2^0 = 32 + 8 + 2 = 42_{10} \end{aligned}$$



42_{10}	$2^5=32$	$42-32=10$ $2^4=16$ $16>10$	$10-8=2$ $2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$
розряди	5	4	3	2	1	0
Цифра числа	1	0	1	0	1	0
	42_{10}	32	0	8	0	2

Переведення дробових чисел з однієї системи числення до іншої

- Для переведення дробу з M -кової системи числення до N -кової потрібно **послідовно множити** дробову частину на основу системи числення N .
- **Цілі частини добутків**, отриманих у результаті виконання послідовності операцій множення, є **цифрами дробу** в N -ковій системі числення.

Переведення дробових чисел з однієї системи числення до іншої

Нехай задане M -кове дробове число V , ($0 \leq V < 1$). Припустимо, що у N -ковій системі числення число V має запис $V = (0, x_{-1}x_{-2}\dots)_N$, тобто

$$V = N^{-1}(x_{-1} + N^{-1}(x_{-2} + \dots)).$$

Цифри x_{-i} невідомі. Помножимо обидві частини рівності на N :

$$V \times N = x_{-1} + N^{-1}(x_{-2} + \dots).$$

Значення цифри x_{-1} отримаємо, взявши цілу частину добутку $V \times N$, тобто

$$x_{-1} = [V \times N], \quad N^{-1}(x_{-2} + \dots) = \{V \times N\},$$

де $[V \times N]$ та $\{V \times N\}$ позначають цілу та дробову частини $V \times N$.

Помножимо дробову частину $\{V \times N\}$ на N і знову візьмемо цілу частину. В результаті отримаємо значення цифри x_{-2} і т. д.

Переведення дробових чисел з однієї системи числення до іншої

$$V = N^{-1}(x_{-1} + N^{-1}(x_{-2} + \dots)).$$

Якщо **число V** задане у вигляді скінченного десяткового дробу і число N має серед своїх простих дільників як 2, так і 5, то зображення числа V в N -ковій системі числення буде **скінченим**.

Інакше в N -ковій системі числення десятковий дріб V може зображуватися **некінченим періодичним дробом**

Переведення дробових чисел з однієї системи числення до іншої

Приклад

Перевести десятковий дріб $0,75_{10}$ до двійкової системи.

Послідовність операцій множення така:

- $0,75 \times 2 = 1,5$; ціла частина **1**,
дробова частина = 0,5;
- $0,5 \times 2 = 1$; ціла частина **1**,
дробова частина = 0.
- Усі подальші цифри будуть нулями, тому $0,11_2$ є скінченим двійковим зображенням дробу $0,75_{10}$.

Приклад 2

У процесі переведення десяткового дробу $0,26_{10}$ у двійкову систему виконується така послідовність операцій множення:

$$0,26 \times 2 = 0,52 \rightarrow 0; \quad (0 - \text{цила частина})$$

$$0,52 \times 2 = 1,04 \rightarrow 1; \quad (1 - \text{цила частина})$$

$$0,04 \times 2 = 0,08 \rightarrow 0;$$

$$0,08 \times 2 = 0,16 \rightarrow 0;$$

$$0,16 \times 2 = 0,32 \rightarrow 0;$$

$$0,32 \times 2 = 0,64 \rightarrow 0,$$

$$0,64 \times 2 = 1,28 \rightarrow 1.$$

Точне двійкове зображення десяткового дробу $0,26$ є нескінченим періодичним, а його наближення $0,010\ 000\ 1_2$.

Приклад

Переведення десяткового дробу 0,8 у **шістнадцяткову** систему числення здійснюється так:

- $0,8 \times 16 = 12,8$ з цілою частиною **12** і дробовою частиною 0,8.
- Десяткове число **12** замінююємо на шістнадцяткове значення **C₁₆**.
- Усі подальші шістнадцяткові цифри будуть також дорівнювати **C**.

Отже, маємо періодичний дріб **0,(C)₁₆**.

Переведення чисел з двійкової до 8-кової та 16-кової системи числення

Оскільки $2^3 = 8$, $2^4 = 16$, то кожні 3 двійкові розряди замінюються одною 8-ковою цифрою, кожні 4 двійкові розряди замінюються одним 16-ковим символом. Відлік розрядів для **цілого** числа здійснюється зліва –направо, для **дробового** числа – справа-наліво

Приклад

$$42_{10} = 101010_2 = 52_8 = 2A_{16}$$

Перевірка

$$52_8 = 5 * 8^1 + 2 * 8^0 = 40 + 2 = 42_{10}$$

$$2A_{16} = 2 * 16^1 + 10 * 16^0 = 32 + 10 = 42_{10}$$



Чому в комп'ютері не використовується десяткова система числення?

Завдання для самостійної роботи

1. Опрацювати матеріал презентації 1 (розділ 1 підручника Ковалюк Т.В. «Алгоритмізація та програмування»)
2. Намалювати структурну схему домашнього комп'ютеру зі специфікацією комплектуючих
3. Натренуватись в переведенні чисел з одної системи числення в інші в ручному режимі (калькулятор використати тільки для перевірки правильності результату)

Контрольні запитання

1. Дайте означення поняття «архітектура комп’ютера».
2. Сформулюйте основні принципи фон Неймана.
3. Які функціональні блоки входять до складу комп’ютера?
4. У чому полягає призначення процесора та його регістрів?
5. Чим відрізняється кеш-пам’ять від інших типів оперативної пам’яті?
6. Що таке машинна команда?
7. Що таке позиційна система числення і як знайти значення числа за його записом у певній позиційній системі?
8. Чому інформація в комп’ютері записується у двійковій системі числення?
9. Для чого використовується шістнадцяткова система числення?
10. Як перевести десяткове число до будь-якої іншої системи числення?
11. Що таке програма?
12. Чим мова програмування високого рівня відрізняється від машинної мови?