

СУЧАСНІ ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ

**122 «Комп'ютерні науки»
КН-18**

2020 / 2021 навчальний рік

ВВЕДЕННЯ до ДИСЦИПЛІНИ

ОБЪЕМ УЧЕБНОЙ РАБОТЫ

Всего 6 кредитов, 180 часов

- **Аудиторные**

- Лекции 32 часа 16 лекции

- Лабораторные 48 часов 16 л.р

- **Курсовой проект (КП)**

- **Экзамен**

ПРОГРАМА. Тематика лекцій

МОДУЛЬ 1. Високорівнева мова програмування Python

МОДУЛЬ 2. Технології програмування

МОДУЛЬ 3. Уніфікована мова моделювання UML

ПРОГРАМА.

Тематика лабораторних занять

- 1 Мова програмування Python.
- 2 Локальна та колективна розробка програм. GitHub.
- 3 Проектування програм. Реінжиніринг програм.

Тематика курсового проектування. Створення програмного пакету

- 1 Алгебраїчні рівняння та системи лінійних рівнянь.
- 2 Інтегрування.
- 3 Звичайні диференціальні рівняння.

ОЦІНЮВАННЯ

ЗАГАЛЬНА ОЦІНКА



КУРСОВИЙ ПРОЕКТ



Башков Євген Олександрович
Д.т.н., проф., кафедра прикладної математики
mail: *eabashkov@i.ua*

1. Написати листа. Вказати ПІБ та групу. В темі завжди писати **2021MPTCS19**
2. На Github <https://github.com> і підключитися до репозитарію **eabshkvprof/2021_Mod_Prog_Techn**

Можна зайти https://github.com/eabshkvprof/2021_Mod_Prog_Techn

Команда MS Teams [2020-2021. КН-19. Технології програмування](#)

МЕТА КУРСУ

отримання знань та навиків, які
необхідні для проектування та
розробки програмних додатків на
основі сучасних технологій
програмування.

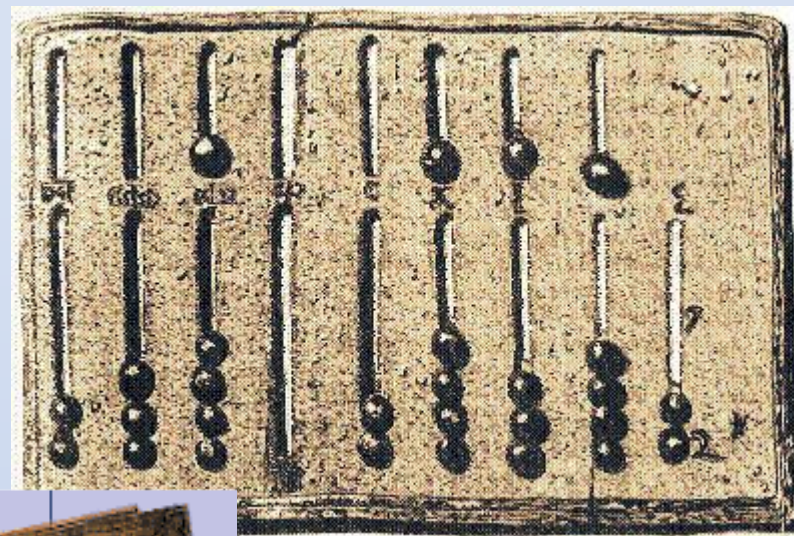
ВВЕДЕННЯ

1. *Історія ОТ.*
2. *Історія програмування.*
3. *Сучасні мови програмування.*
Класифікація
4. *Python*

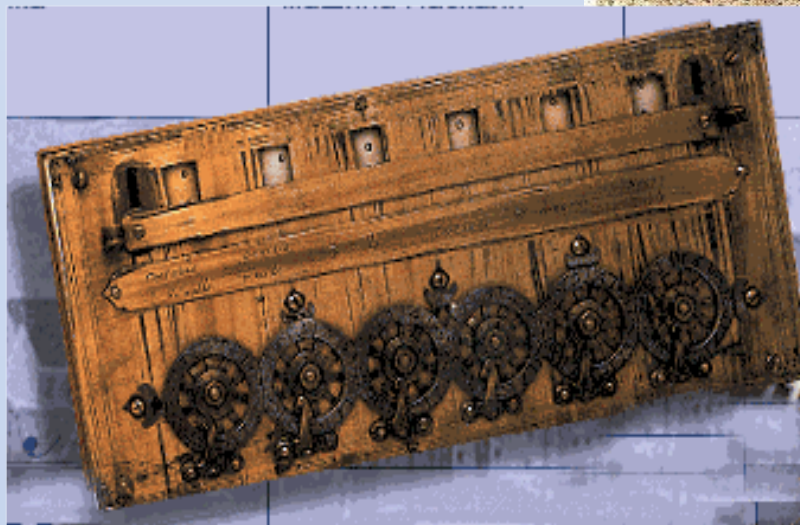
ВВЕДЕННЯ. Історія ОТ

Комп'ютер (обчислювальна машина, ЕОМ, ЦОМ) – *програмно* - керований пристрій для обробки інформації.

Абак (?)

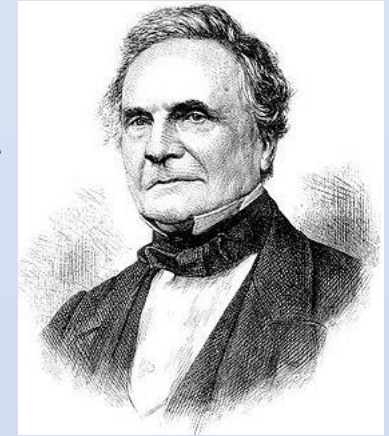
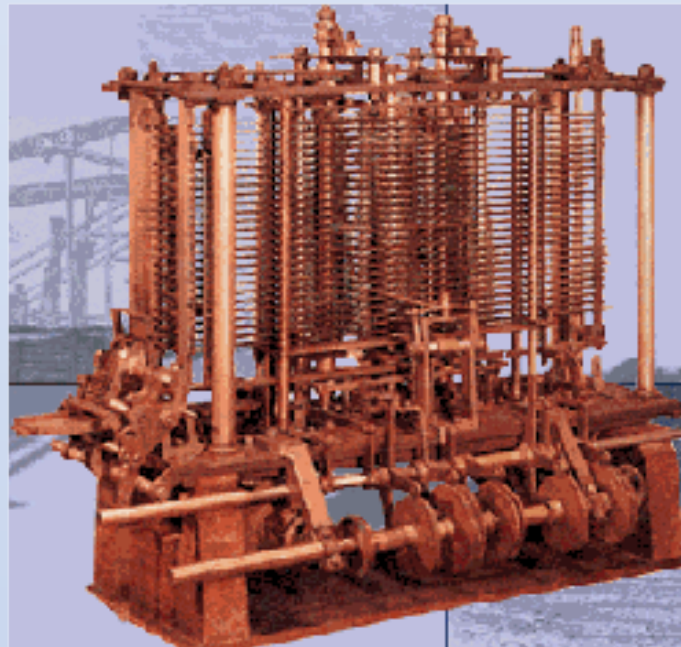
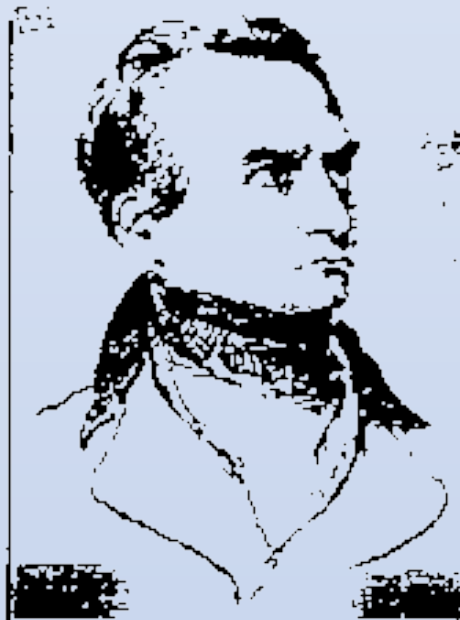


Блез Паскаль (1623- 1662)



ВВЕДЕННЯ. Історія ОТ

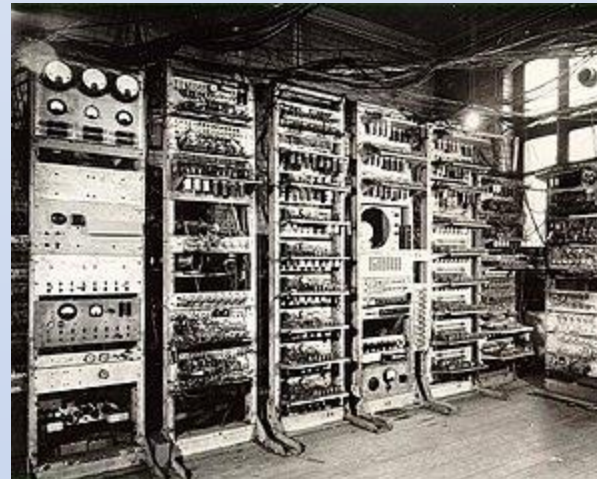
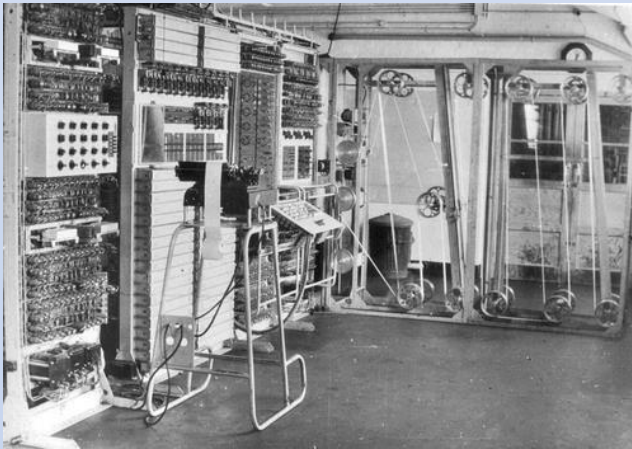
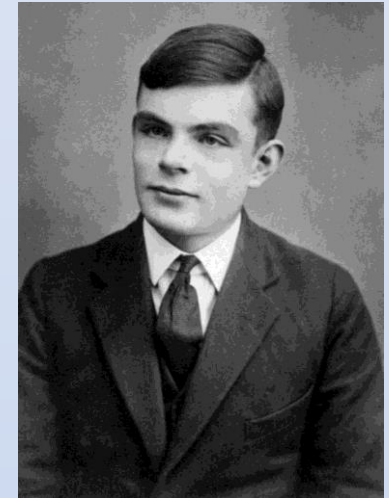
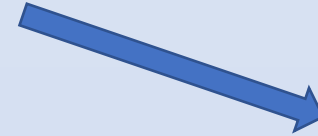
Чарльз Беббідж (1792-1881)



ВВЕДЕННЯ. Історія ОТ

Алан Тьюрінг (1912- 1954)

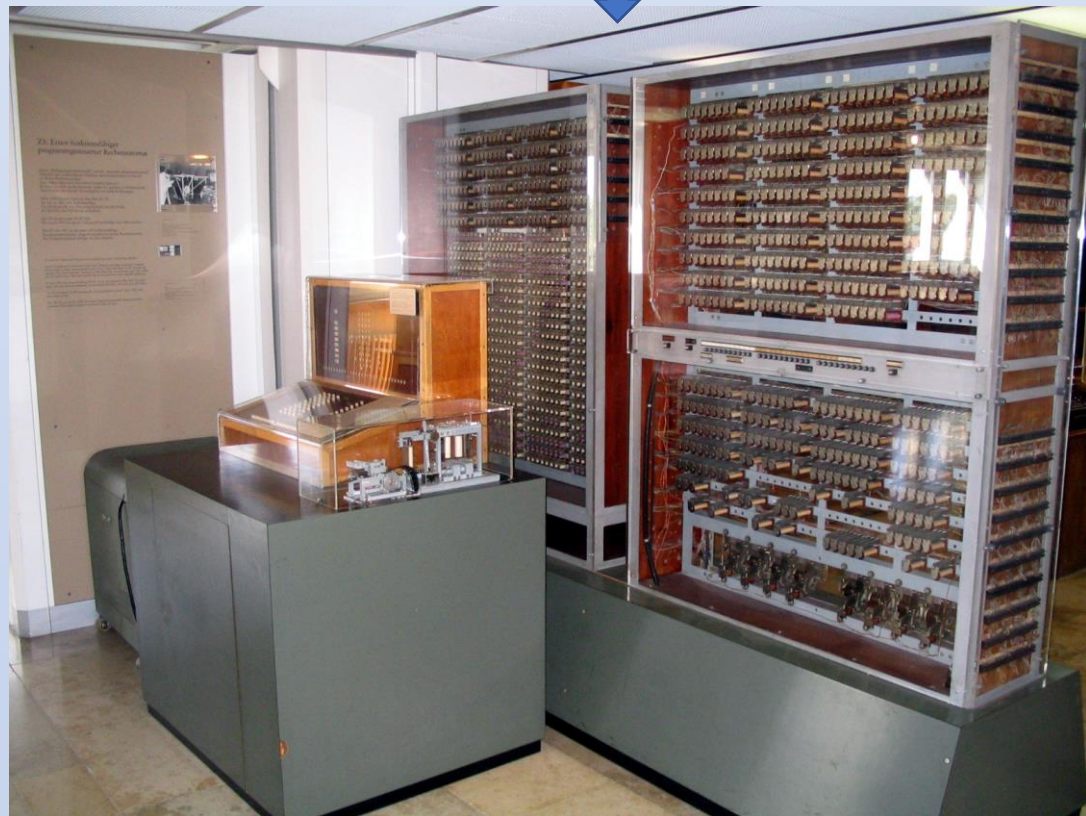
Colossus, Mark



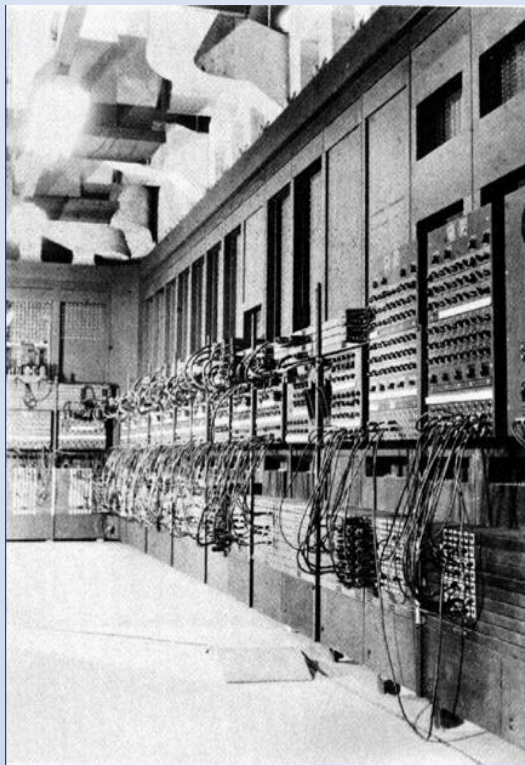
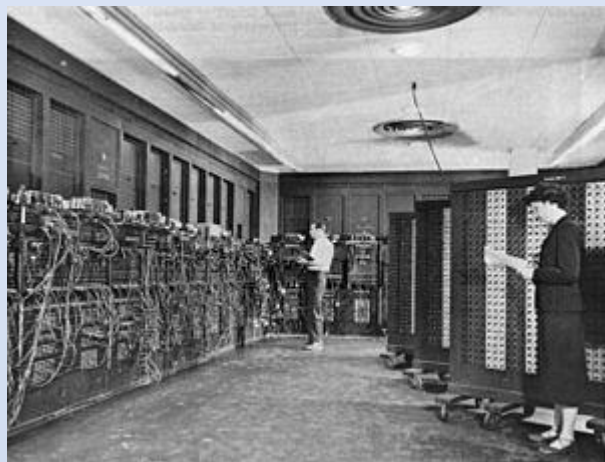
ВВЕДЕННЯ. Історія ОТ

Конрад Цузе (1910- 1995)

Z1 (1938), Z2, Z3



ВВЕДЕННЯ. Історія ОТ



Джон фон Нейман (1903-1957

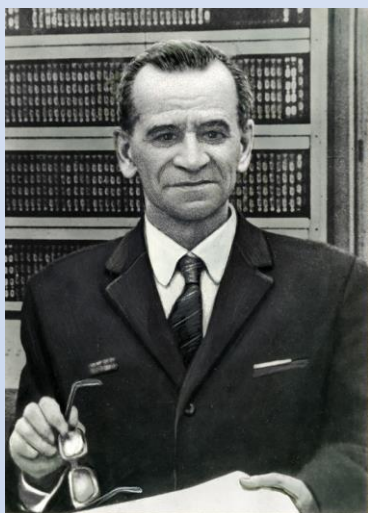
Говард Ейкен (Harvard Mark I, II, III)

Джон Еккерт, Джон Моклі (ENIAC)

ВВЕДЕННЯ. Історія ОТ

Сергій Лебедєв (1902-1974)

МЭСМ, БЭСМ



Історія ОТ дивись <https://uk.wikipedia.org/wiki/компютер>

<http://ua.uacomputing.com/stories/mesm/>

ВВЕДЕННЯ. Історія програмування

Мова програмування — штучна формальна система, засобами якої можна виражати алгоритми. Мова програмування визначає набір лексичних, синтаксичних, та семантичних правил, що задають зовнішній вигляд програми і дії, які виконує виконавець (комп'ютер) під її управлінням

ВВЕДЕННЯ. Історія програмування

Жозеф Жаккард (1752-1843)



| 1 2 3 4 | | | | | | | | | | | | | | | 5 6 7 | | | | | | | | | | | | | | | |
|---------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

ВВЕДЕННЯ. Історія програмування

Ада Лавлейс (1815-1852)



Sketch of *The Analytical Engine* Invented by Charles Babbage

By L. F. MENABREA
of Turin, Officer of the Military Engineers

from the *Bibliothèque Universelle de Genève*, October, 1842, No. 82

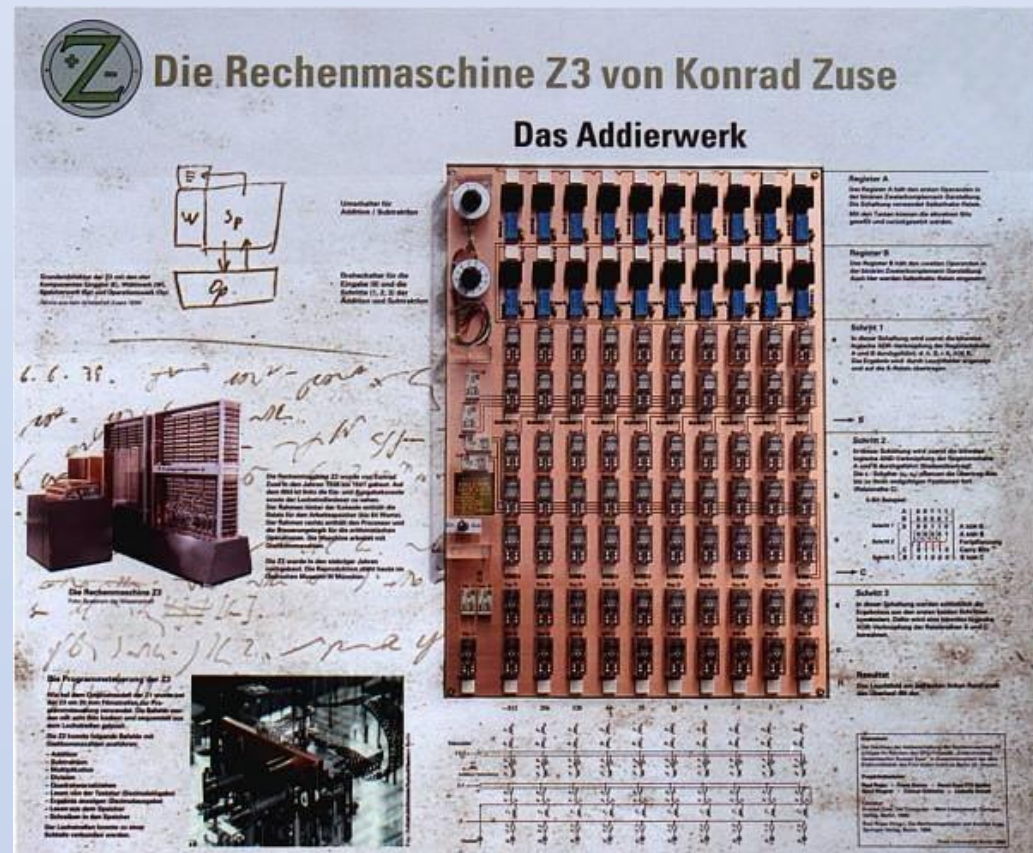
With notes upon the Memoir by the Translator
ADA AUGUSTA, COUNTESS OF LOVELACE

| Columns on which are inscribed the primitive data | Number of the operations | Cards of the operations | | Variable cards | | | Statement of results |
|---|--------------------------|----------------------------|--------------------------|---|---|--|--|
| | | No. of the Operation-cards | Nature of each operation | Columns acted on by each operation | Columns that receive the result of each operation | Indication of change of value on any column | |
| ${}^1V_0 = m$ | 1 | 1 | \times | ${}^1V_0 \times {}^1V_4 = {}^1V_6$ | 1V_6 | $\left\{ \begin{array}{l} {}^1V_0 = {}^1V_0 \\ {}^1V_4 = {}^1V_4 \end{array} \right\}$ | ${}^1V_6 = mn'$ |
| ${}^1V_1 = n$ | 2 | " | \times | ${}^1V_3 \times {}^1V_1 = {}^1V_7$ | 1V_7 | $\left\{ \begin{array}{l} {}^1V_3 = {}^1V_3 \\ {}^1V_1 = {}^1V_1 \end{array} \right\}$ | ${}^1V_7 = m'n$ |
| ${}^1V_2 = d$ | 3 | " | \times | ${}^1V_2 \times {}^1V_4 = {}^1V_8$ | 1V_8 | $\left\{ \begin{array}{l} {}^1V_2 = {}^1V_2 \\ {}^1V_4 = {}^0V_4 \end{array} \right\}$ | ${}^1V_8 = dn'$ |
| ${}^1V_3 = m'$ | 4 | " | \times | ${}^1V_5 \times {}^1V_1 = {}^1V_9$ | 1V_9 | $\left\{ \begin{array}{l} {}^1V_5 = {}^1V_5 \\ {}^1V_1 = {}^0V_1 \end{array} \right\}$ | ${}^1V_9 = d'n$ |
| ${}^1V_4 = n'$ | 5 | " | \times | ${}^1V_0 \times {}^1V_5 = {}^1V_{10}$ | ${}^1V_{10}$ | $\left\{ \begin{array}{l} {}^1V_0 = {}^0V_0 \\ {}^1V_5 = {}^0V_5 \end{array} \right\}$ | ${}^1V_{10} = d'm$ |
| ${}^1V_5 = d'$ | 6 | " | \times | ${}^1V_2 \times {}^1V_3 = {}^1V_{11}$ | ${}^1V_{11}$ | $\left\{ \begin{array}{l} {}^1V_2 = {}^0V_2 \\ {}^1V_3 = {}^0V_3 \end{array} \right\}$ | ${}^1V_{11} = dm'$ |
| | 7 | 2 | — | ${}^1V_6 - {}^1V_7 = {}^1V_{12}$ | ${}^1V_{12}$ | $\left\{ \begin{array}{l} {}^1V_6 = {}^0V_6 \\ {}^1V_7 = {}^0V_7 \end{array} \right\}$ | ${}^1V_{12} = mn' - m'n$ |
| | 8 | " | — | ${}^1V_8 - {}^1V_9 = {}^1V_{13}$ | ${}^1V_{13}$ | $\left\{ \begin{array}{l} {}^1V_8 = {}^0V_8 \\ {}^1V_9 = {}^0V_9 \end{array} \right\}$ | ${}^1V_{13} = dn' - d'n$ |
| | 9 | " | — | ${}^1V_{10} - {}^1V_{11} = {}^1V_{14}$ | ${}^1V_{14}$ | $\left\{ \begin{array}{l} {}^1V_{10} = {}^0V_{10} \\ {}^1V_{11} = {}^0V_{11} \end{array} \right\}$ | ${}^1V_{14} = d'm - dm'$ |
| | 10 | 3 | \div | ${}^1V_{13} \div {}^1V_{12} = {}^1V_{15}$ | ${}^1V_{15}$ | $\left\{ \begin{array}{l} {}^1V_{13} = {}^0V_{13} \\ {}^1V_{12} = {}^1V_{12} \end{array} \right\}$ | ${}^1V_{15} = \frac{dn' - d'n}{mn' - m'n} =$ |
| | 11 | " | \div | ${}^1V_{14} \div {}^1V_{12} = {}^1V_{16}$ | ${}^1V_{16}$ | $\left\{ \begin{array}{l} {}^1V_{14} = {}^0V_{14} \\ {}^1V_{12} = {}^0V_{12} \end{array} \right\}$ | ${}^1V_{16} = \frac{d'm - dm'}{mn' - m'n} =$ |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

ВВЕДЕННЯ. Історія програмування

Конрад Цузе (1945)

мова PlanKaKuI - обчислювач планів



ВВЕДЕННЯ. Історія програмування

Тоні Брукнер (1951). Manchester Mark I - Мова Autocode



An Example

Tabulate Sievert's integral $\int_0^y e^{-a \sec x} dx$ for $y = 1(1)90^\circ$ and particular values of a . The method adopted is to tabulate the integrand for $x = 0(\frac{1}{2}) 90^\circ$ and calculate the integral step-by-step using Simpson's rule to evaluate the increments;

$$\int_0^{y+h} = \int_0^y + \frac{h}{6} \left[f(y) + 4f\left(y + \frac{h}{2}\right) + f(y+h) \right]$$

where $h = \frac{\pi}{360}$

```

      f → 180
      h = π/360                                (5 m
      s = 0(1)179                               1
      b = φ cos (sh)                             (23 m
      fs = φ exp (-a/b)                         (23 m
      repeat                                     4
      f180 = 0                                   4
      y = 0                                       4
      r = 1(1)90                                 1
      newline                                   (90 m
      print (r)2,0                               (150 m
      space                                     (60 m
      s = 2r - 1                                 19
      f = f(s-1) + 4fs + f(s+1)                22
      y = y + hf/3                               (5 m
      print (y) 1,6                             (333 m
      repeat                                     4
      end
```

ВВЕДЕННЯ. Історія програмування

Джон Бекус (1954).

Мова FORTRAN (!!! Fortran 2018)

.....

1958 LISP

1958 ALGOL

1959 COBOL

1964 BASIC

1971 PASCAL

1972 C

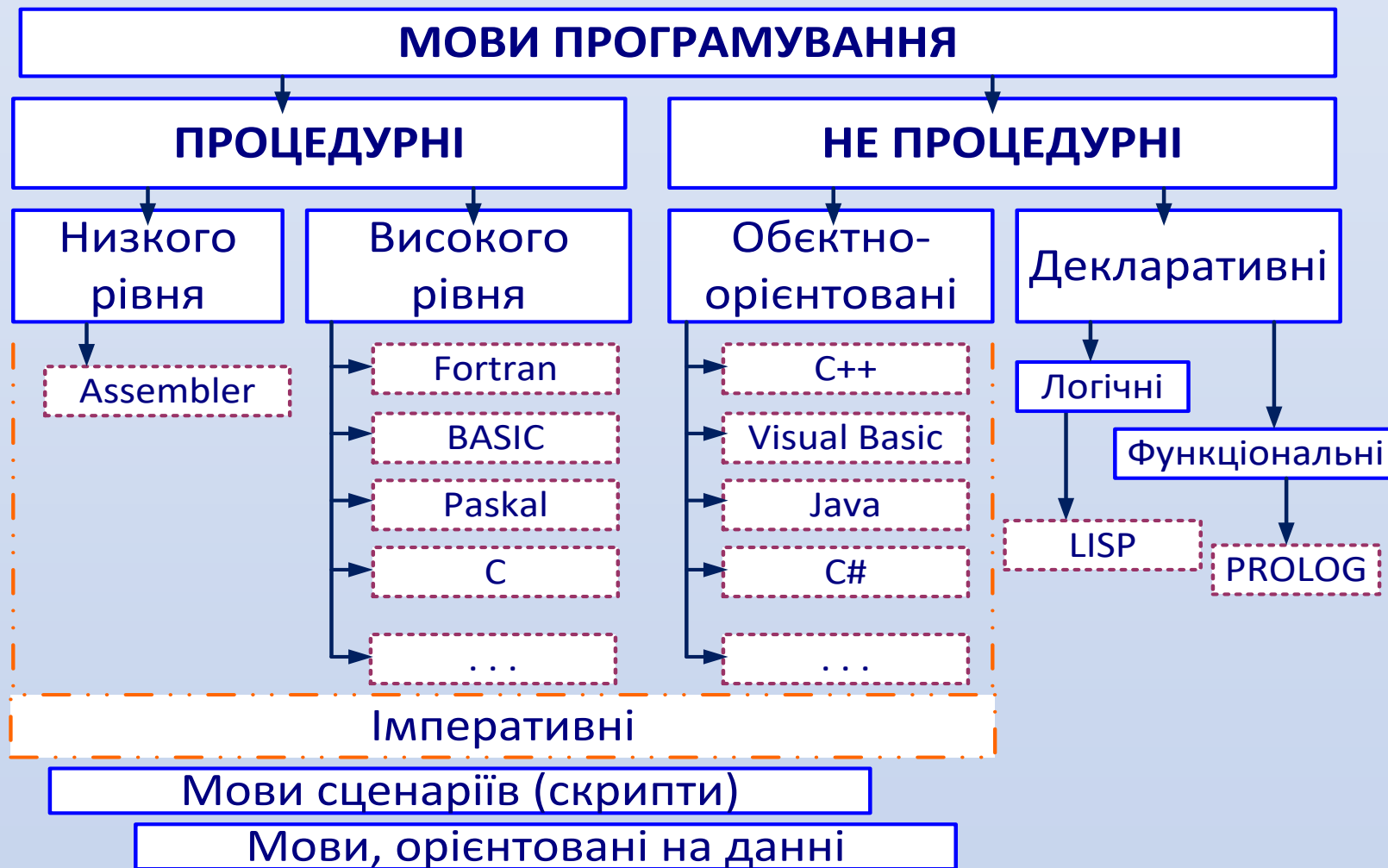
Історія програмування дивись:

https://uk.wikipedia.org/wiki/мова_програмування

https://ru.wikipedia.org/wiki/хронология_языков_программирования

ВВЕДЕННЯ. Мови програмування

Класифікація (спрощена !!!)



! Більш ніж **2000** мов програмування

ВВЕДЕННЯ. Мови програмування

Процедурні мови - мови високого рівня, в яких використовується метод розбиття програм на окремі пов'язані між собою модулі - підпрограми (процедури і функції). Компоненти мови складаються з послідовності операторів, які використовують бібліотечні процедури і функції.

Непроцедурні мови – мови високого рівня орієнтовані на використання «технологічне» програмування – маніпулювання деяким сутностями предметної області.

ВВЕДЕННЯ. Мови програмування

Мови низького рівня – орієнтовані на певний тип процесора, враховують його архітектурні особливості (програмування в кодах, **assembler, macroassembler**).

Мови високого рівня – не враховують особливості конкретного процесора, програми на мовах високого рівня достатньо легко переносяться з одної архітектури на іншу.

ВВЕДЕННЯ. Мови програмування

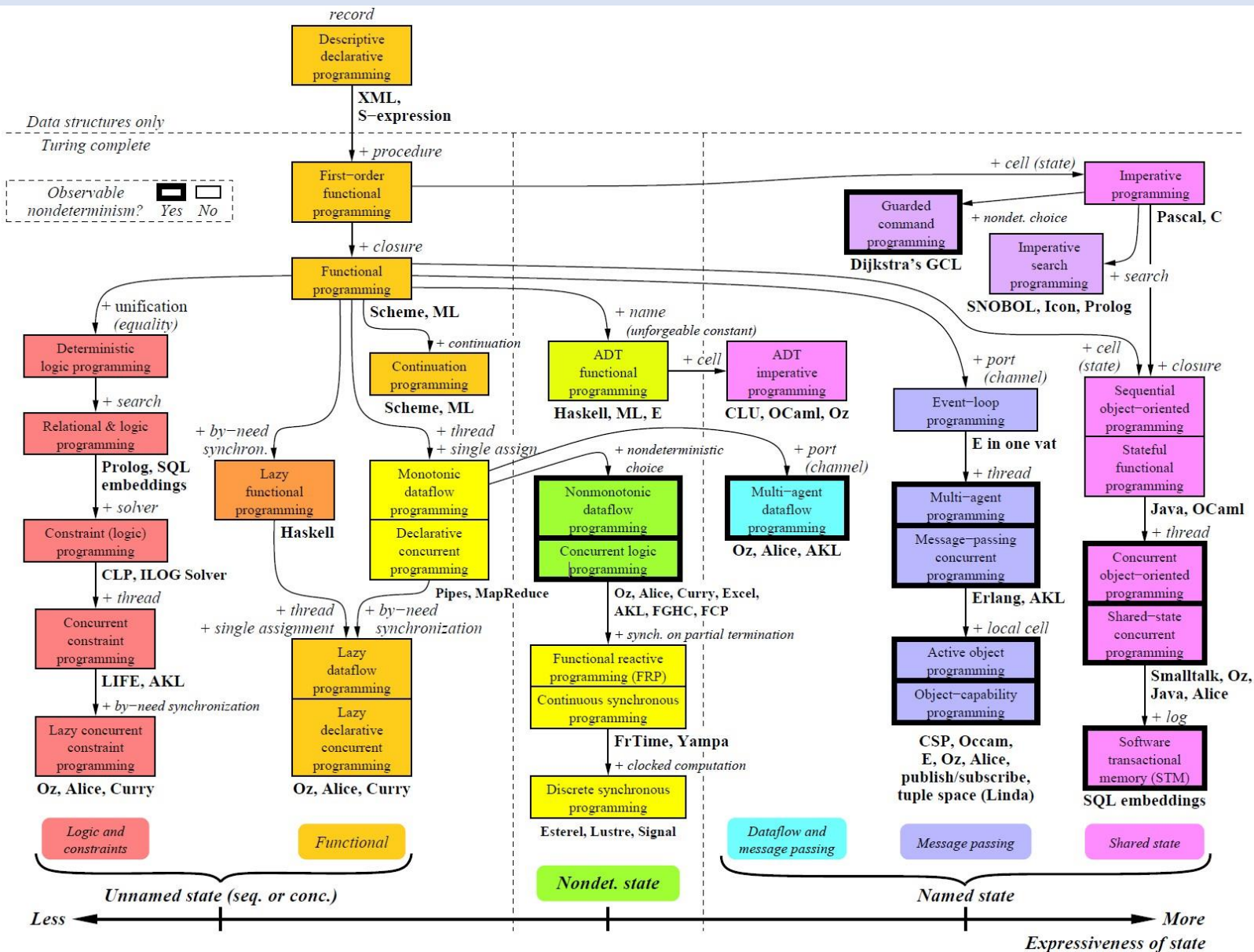
Об'єктно - орієнтовні мови –

використовують множину програмних об'єктів – сутностей, що об'єднують в собі дані (поля) та дії (методи) (C++, Python, ...).

Декларативні мови – описують результат, який потрібно отримати, замість послідовності операцій з отримання цього результату (SQL, HTML).

Імперативні мови – детально описують деякий алгоритм отримання результатів.

ВВЕДЕННЯ. Мови програмування



ВВЕДЕНИЯ. Мови програмування

Реально використовується $\cong 100$ мов

| | C | C++ | C# | Java | Python | Delphi | Ruby | PHP | Small talk | Lisp |
|----------------|---|-----|----|------|--------|--------|------|-----|------------|------|
| Императивные | + | + | + | + | + | + | + | + | + | + |
| Декларативные | - | - | - | - | + | - | + | - | + | + |
| Функциональные | - | - | - | - | + | - | + | - | + | + |
| Объектные | - | + | + | + | + | + | + | + | + | + |
| Логические | - | - | - | - | - | - | - | - | + | + |
| Распределенные | - | - | - | - | - | - | - | - | + | + |

Дивись:

<https://www.youtube.com/watch?v=Og847HVwRSI&feature=youtu.be>

PYTHON #1

1. Ідентифікатори
2. Запис програм
3. Змінні, типи, типізація
4. Типи даних: числа
5. Оператори, операції
6. Логічний тип даних, логічні вирази
7. Алгоритмічні структури:
слідкування, розгалуження, цикл

PYTHON #1

Python (Пайтон, 1990) - інтерпретована, об'єктно - орієнтовна мова програмування високого рівня .

Python підтримує модулі та пакети. Інтерпретатор **Python** та стандартні бібліотеки доступні на всіх основних платформах.



Python підтримується : об'єктно – орієнтоване, процедурне, функціональне та аспектно-орієнтовне програмування.

Офіційний сайт : <https://www.python.org/>

PYTHON #1



Розвиток:

Python 1.0 (1980)

Python 2.0 (2000)

Python 3.0 (2008) // зворотне несумісний

Anaconda (MIT) – open source дистрибутив

- **Spyder**
- **JupyterLab**
- **Jupyter Notebook**

<https://anaconda.org/>

PYTHON #1

Переваги:

- Інтерпретована мова програмування.
- Динамічна типізація.
- Модульність.
- Вбудована підтримка Unicode.
- Об'єктно - орієнтоване програмування.
- Автоматична збірка мусору.
- Інтеграція з C / C++.
- Кросплатформеність.

Недоліки:

- Інтерпретована мова програмування.

ІДЕНТИФІКАТОРИ

Ідентифікатор – деяке ім'я, яке використовується для ідентифікації об'єкту: змінної, функції, класу, модуля ...

Ідентифікатор може містити тільки символи:

- Літери в нижньому регістрі *a ... z*
- Літери в верхньому регістрі *A ... Z*
- Цифри *0 ... 9*
- Нижнє підкреслення *_*

!! Не може починатися з цифри

!! Не може співпадати з зарезервованими словами

ЗАПИС ПРОГРАМИ

!!! Блоки коду відокремлюються за допомогою рядкового відступу

if s>0 :

print ("YES")

else :

print ("NO")

!!! Жорстка вимога

Коментар – фрагмент тексту програми, що ігнорується при виконанні.

Коментар в рядку починаються з символу **#**

I = 0 #index initialization

ЗМІННІ, ТИПИ, ТИПІЗАЦІЯ

Змінна - це ім'я, яке посилається на деякий об'єкт (значення) в пам'яті комп'ютера.

A=3



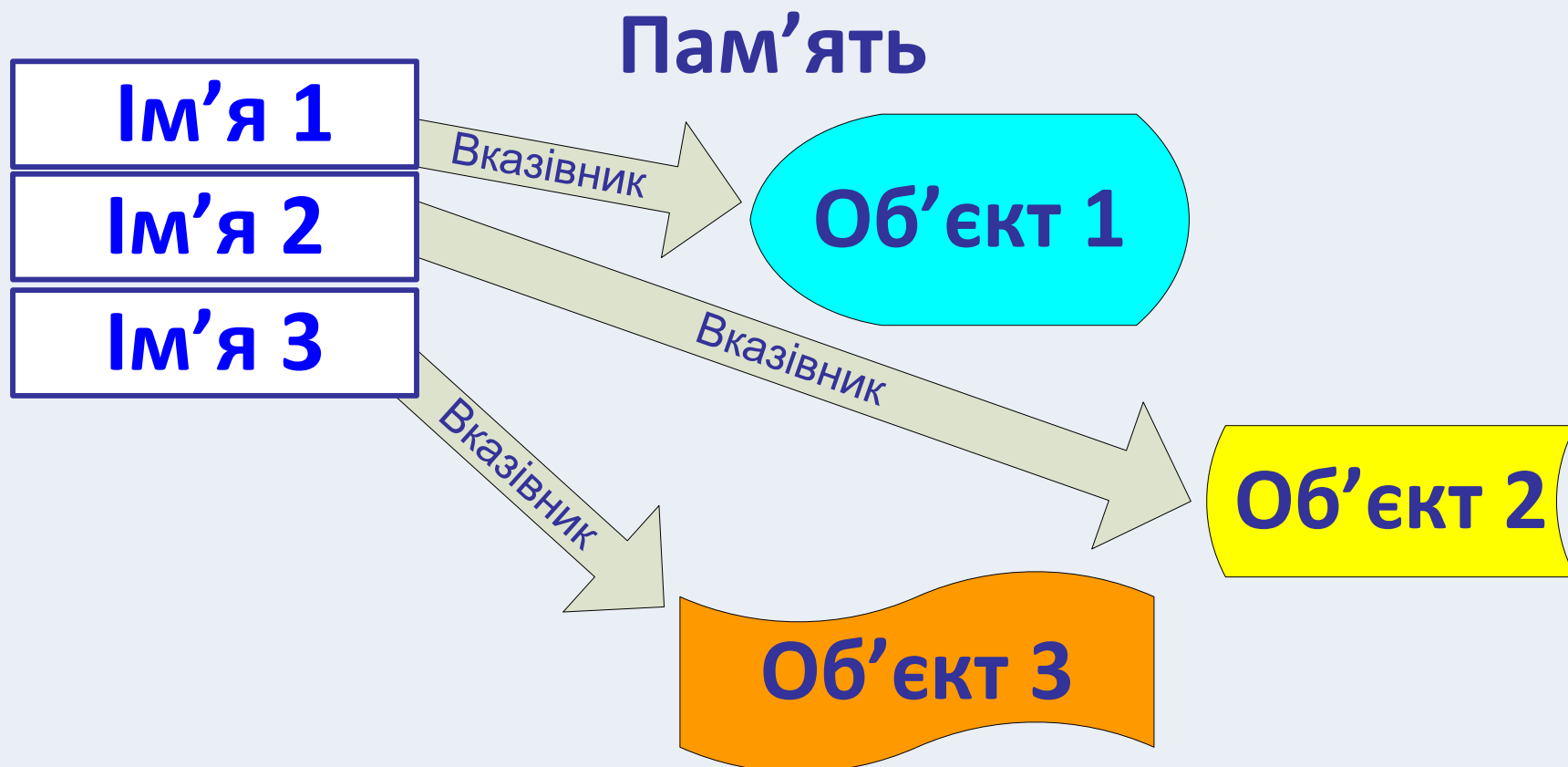
Змінні **створюються** при виконанні операції присвоювання значення.

Для використання в виразах змінна повинна мати значення (**ініціалізована !**).

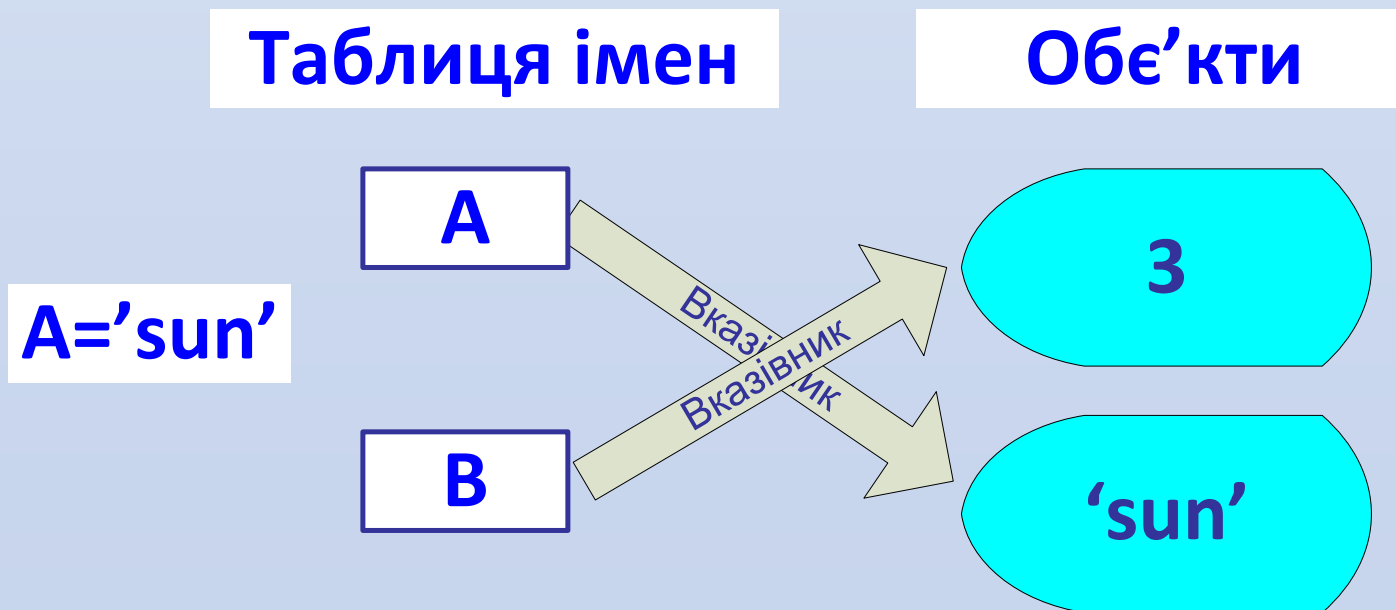
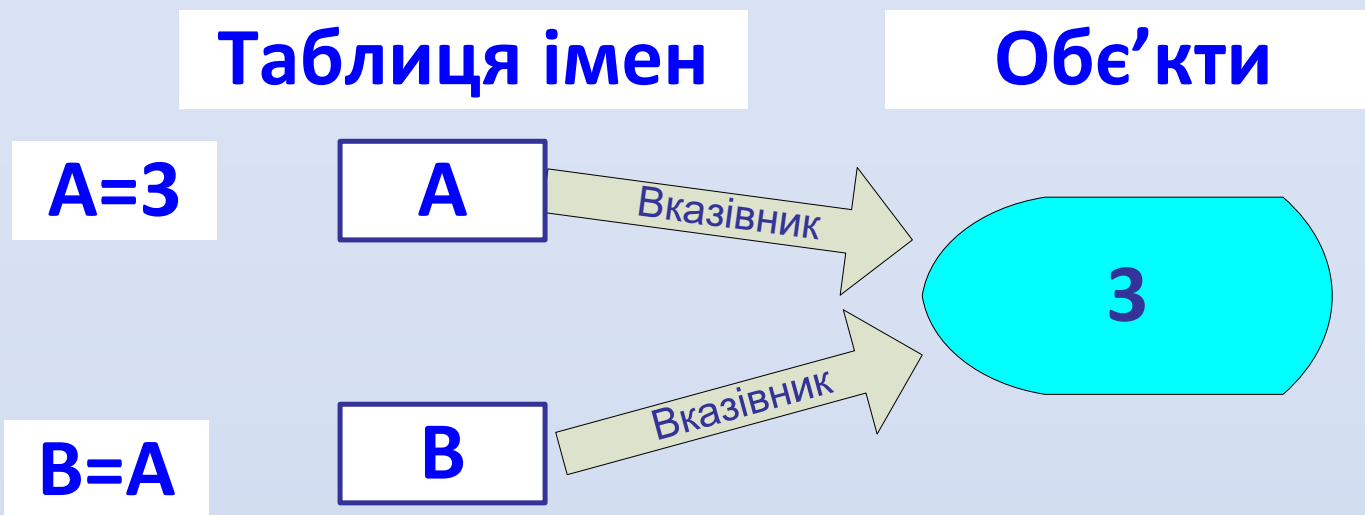
Під час обчислення значень деякого виразу ім'я змінної **заміщується** її значенням.

ЗМІННІ, ТИПИ, ТИПІЗАЦІЯ

!!! Змінна = Ім'я



ЗМІННІ, ТИПИ, ТИПІЗАЦІЯ



ЗМІННІ, ТИПИ, ТИПІЗАЦІЯ

Тип об'єкту – множина значень та множина операцій на цих значеннях. Тип визначає можливі значення та їх сенс, способи зберігання цих значень, можливі операції над значеннями.

Типізація – операція визначення типу інформаційній сутності (об'єкту)

| | СТАТИЧНА | ДИНАМІЧНА |
|--------|----------|------------|
| СИЛЬНА | C# | Python |
| СЛАБКА | C | JavaScript |

ЗМІННІ, ТИПИ, ТИПІЗАЦІЯ

Статична – змінна не може змінити тип.
Визначається на етапі компіляції.

Динамічна – змінна **може** змінити тип.
Визначається при призначенні їй
значення.

Сильна – не допускає виконання
операції при несумісності типів.

Слабка – дозволяє виконання операції
при несумісності типів. Результат ????

| | СТАТИЧНА | ДИНАМІЧНА |
|--------|----------|---------------|
| СИЛЬНА | C# | Python |
| СЛАБКА | C | JavaScript |

ТИП: ЧИСЛО

ОБЄ'КТ --> ЧИСЛО

ЦІЛІ

int

Long int

bool

ДІЙСНІ - *float*


КОМПЛЕКСНІ - *complex*

None

| | ЛИТЕРАЛ |
|----------|------------------------------|
| Int | 123 -24 0 |
| Long int | 999999999999999999L |
| Bool | True False |
| Float | 1.23 -123.45 -32.5E-21 |
| Complex | 3+4j 3.0 +3.0j 4.0j |

python 3.0 int = long int (всe long int)

ЧИСЛА. ОПЕРАТОРИ / ОПЕРАЦІЇ



| Оператор | Опис |
|----------------------|------------------------------------|
| +, - | Унарні +, -. Додавання, віднімання |
| *, / | Множення, Ділення |
| //, % | Цілочисельне ділення, Залишок |
| ** | Піднесення до степеню |
| <, <=, ==, >=, >, != | Зрівняння |
| , &, ^ | Логічні OR, AND, XOR |
| <<, >> | Зсув |

Старшинство операцій = старші виконуються поперед молодших

Тип результату при змішаних операціях - ранжування типів

int → float → complex

ЛОГІЧНІ ВИРАЗИ

Логічна змінна – підклас цілочисельного типу *int*, яка приймає значення 0 або 1 і представлена під час виводу як *False* та *True*

| Оператор | Опис |
|----------------------|--------------------------------------|
| <, <=, ==, >=, >, != | Зрівняння → результат <i>boolean</i> |
| , &, ^ | Логічні OR, AND, XOR |

В більш широкому сенсі:

Кожний об'єкт може бути *True* або *False*.

Об'єкт *True* якщо він не порожній (спрощено не 0)

Об'єкт *False* якщо він порожній (**None**, аналог **NULL**)

СЛІДУВАННЯ

СЛІДУВАННЯ – команди (інструкції) виконуються послідовно одна за іншою.

!!! Кінець рядка - кінець інструкції ~~;~~

!!! Відсутні дужки блоків ~~*begin end { }*~~

!!! Відступи

Допускається декілька інструкцій в один рядок

A = 2; B = 3.25; c = 'kajfhad'

Допускається одна інструкція в декілька рядків // необхідно взяти в дужки
() або []

РОЗГАЛУЖЕННЯ IF

РОЗГАЛУЖЕННЯ – перевірка умови (умов) і виконання відповідного блоку інструкцій

```
if <test 1> :    # умова 1  
    <statements 1> # блок інструкцій 1  
elif <test 2> : # умова 2  
    <statements 2> # блок інструкцій 2  
else :  
    <statements 3> # блок інструкцій 3
```

ЦИКЛ WHILE

ЦИКЛ – структура, що виконує блок інструкцій доки діє деяка умова.

```
while <test> :      # умова  
    <statements 1> # блок інструкцій 1
```

Можливо додатково

```
else :                # необов'язково  
    <statements 2> # блок інструкцій 2
```

Додаткові інструкції (тільки в блоці 1)

```
break      # вихід з циклу  
continue   # перехід до початку циклу  
pass       # пуста інструкція
```

ЦИКЛ FOR

for *<target>* *in* *<object>* : # змінна & умова
 <statements> # блок інструкцій

Можливо додатково

else : # необов'язково
 <statements 2> # блок інструкцій 2

Додаткові інструкції (тільки в блоці)

break # вихід з циклу
continue # перехід до початку циклу
pass # пуста інструкція

Контрольні запитання

- Надайте визначення мови програмування
- Надайте визначення імперативної мови програмування. Наведіть приклади.
- Надайте визначення декларативної мови програмування. Наведіть приклади.
- Надайте визначення мови програмування низького рівня. Наведіть приклади.
- Надайте визначення мови програмування високого рівня. Наведіть приклади.
- Надайте визначення об'єктно - орієнтованої мови програмування Наведіть приклади.
- Надайте базові властивості мови програмування Python.

Контрольні запитання

- Надайте визначення змінної в мові Python. Поясніть, як визначається тип змінної в мові Python.
- Опишіть властивості змінних типу *int* та визначте операції з ними.
- Опишіть властивості змінних типу *float* та визначте операції з ними.
- Опишіть властивості змінних типу *complex* та визначте операції з ними.
- Опишіть властивості змінних типу *bool* та визначте операції з ними.
- Надайте визначення структури *if ... elif ... else* та наведіть приклади використання
- Надайте визначення структури *while ...* та наведіть приклади використання
- Надайте визначення структури *for ... in ...* та наведіть приклади використання

Рекомендована ЛІТЕРАТУРА

- **Малинвський Б.М.** Зберігати довічно. — Київ : Видавничий дім «Києво-Могилянська академія», 2007. — 176 с.
- **Малинвський Б.М.** Нариси з історії комп'ютерної науки та техніки в Україні. — Київ : Феникс, 1998. - 452 с.
- **Апокин И.А., Майстров Л.Е.** Развитие вычислительных машин. М.: «Наука», 1974
- **Апокин И.А., Майстров Л.Е., Эдлин И.С.** Чарльз Бэббидж. М.: «Наука», 1981
- **Апокин И.А., Майстров Л.Е.** История вычислительной техники. М.: «Наука», 1990

Рекомендована ЛІТЕРАТУРА

- **Програмування числових методів мовою Python:** підруч. / А. В. Анісімов, А. Ю. Дорошенко, С. Д. Погорілий, Я. Ю. Дорогий ; за ред. А. В. Анісімова. – К. : Видавничо-поліграфічний центр "Київський університет", 2014. – 640 с.
- **Програмування числових методів мовою Python:** навч. посіб. / А. Ю. Дорошенко, С. Д. Погорілий, Я. Ю. Дорогий, Є. В. Глушко ; за ред. А. В. Анісімова. – К. : Видавничо-поліграфічний центр "Київський університет", 2013. – 463 с.
- **Основи програмування Python:** Підручник для студ. спеціальності 122 «Компютерні науки» / А.В.Яковенко; КПІ.- Київ: КПІ, 2018 . – 195 с.
- **Лутц М. Изучаем Python, 4-е издание. - СПб.: Символ-Плюс. 2011.- 1280 с.: ил.**

The END
Mod 1. Lec 1.