

СУЧАСНІ ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ

**122 «Комп'ютерні науки»
КН-18**

2019 / 2020 навчальний рік

ОБЪЕМ УЧЕБНОЙ РАБОТЫ

Всего 6 кредитов, 180 часов

- **Аудиторные**

- Лекции 48 часов 24 лекции

- Лабораторные 48 часов 16 лаб.работ

- **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 Евгений Александрович
Д.т.н., проф., кафедра прикладной
математики

mail: *eabashkov@i.ua*

1. Написать письмо, указать полностью ФИО
и группу. !!!! В теме письма ВСЕГДА писать
Стационар **2020MPTCS18**

2. Зарегистрироваться на Github
<https://github.com> и подключиться к
репозитарию
eabshkvprof/2020_Mod_Prog_Tech

Александров Никита Александрович,
асс. Каф. ПМИ

МЕТА КУРСУ

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

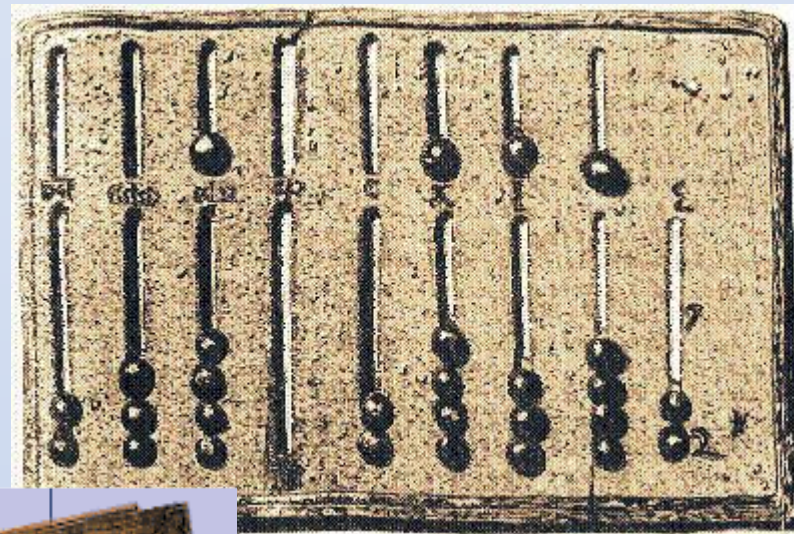
ВВЕДЕННЯ

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

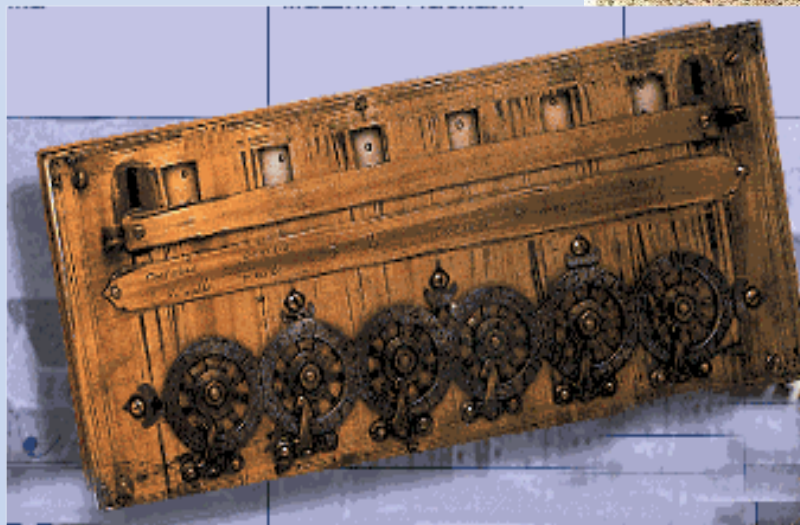
ВВЕДЕННЯ. Історія ОТ

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

Абак (?)

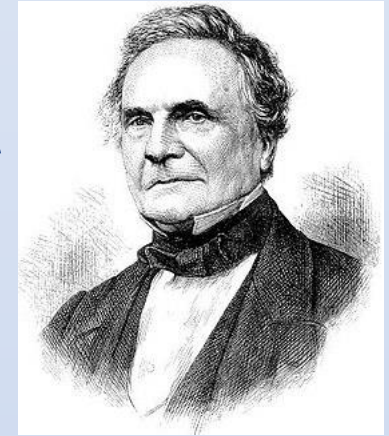
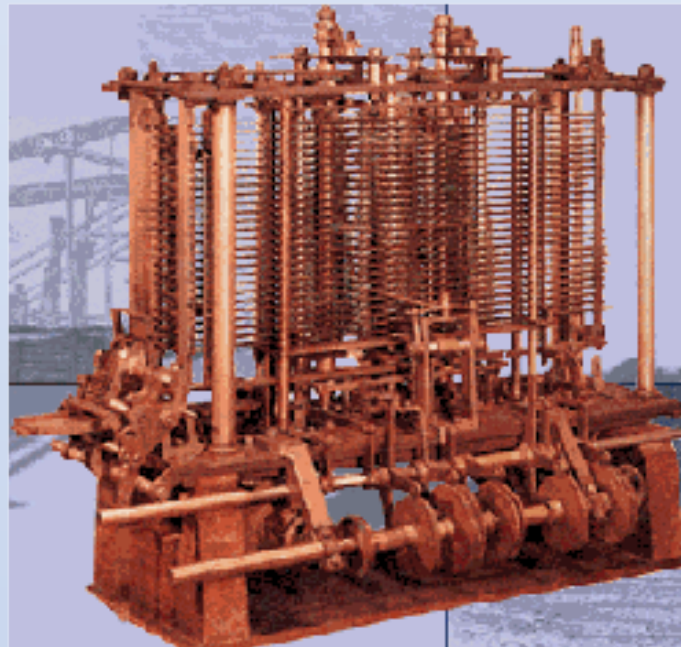
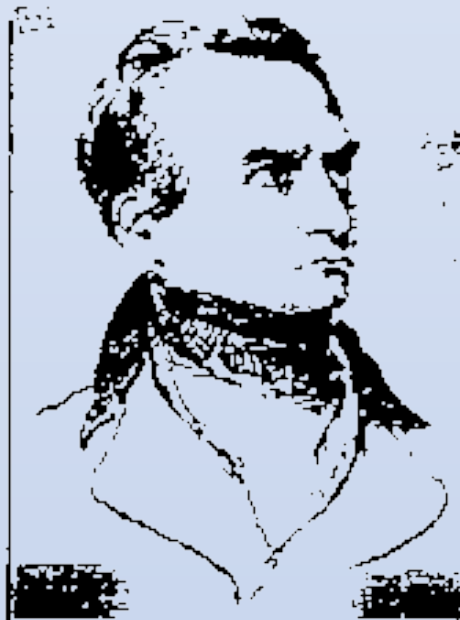


Блез Паскаль (1623- 1662)



ВВЕДЕННЯ. Історія ОТ

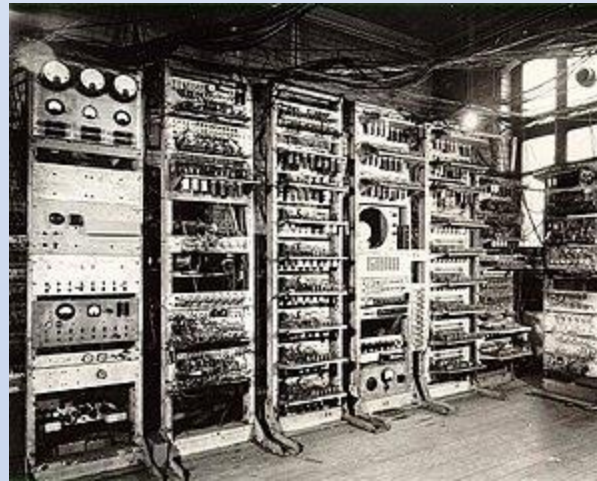
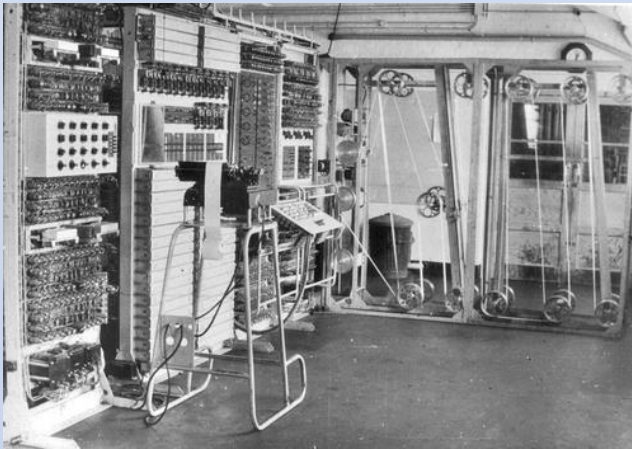
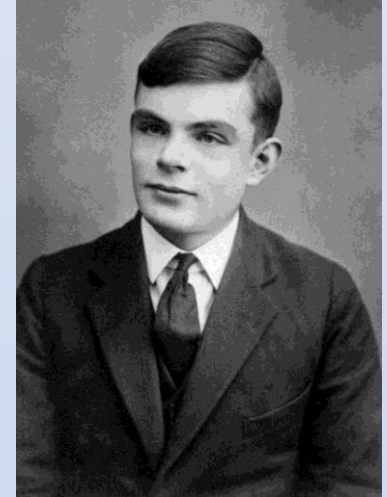
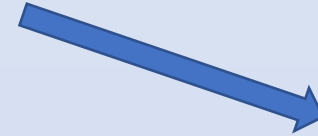
Чарльз Беббідж (1792-1881)



ВВЕДЕННЯ. Історія ОТ

Алан Тьюрінг (1912- 1954)

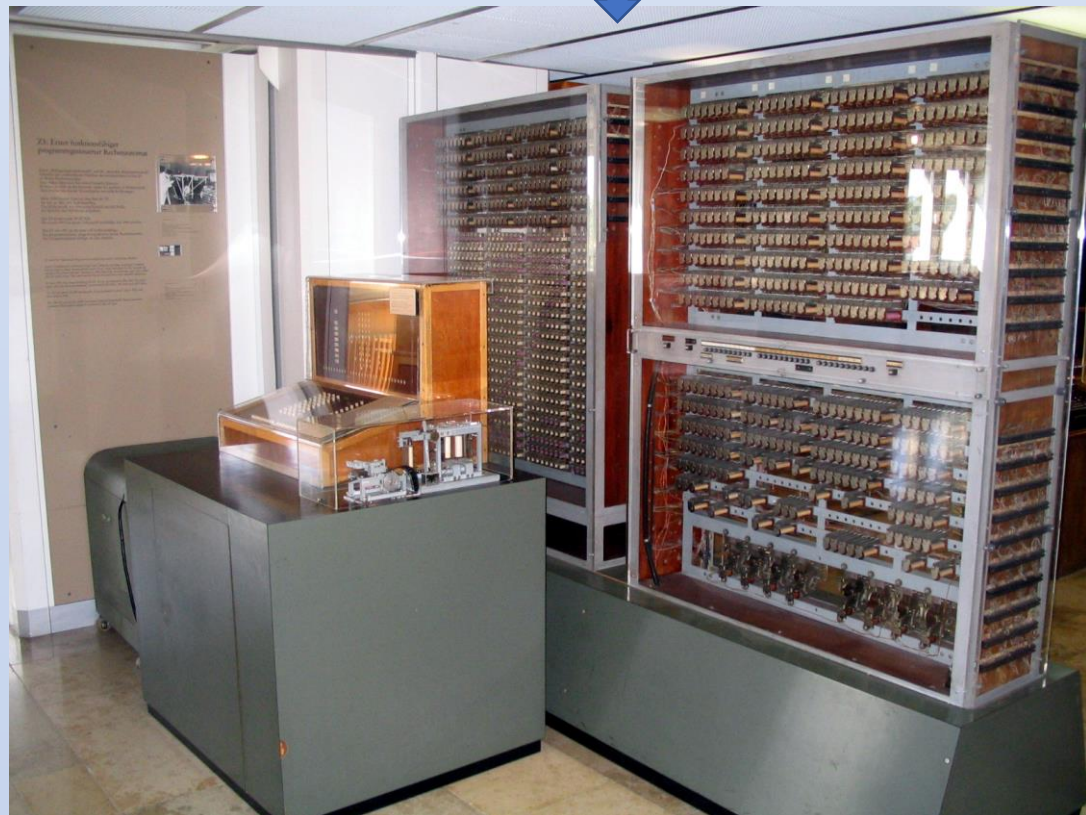
Colossus, Mark



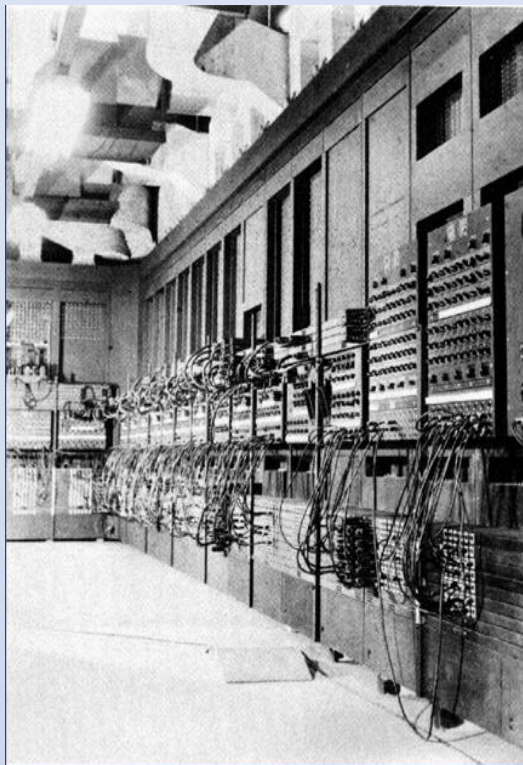
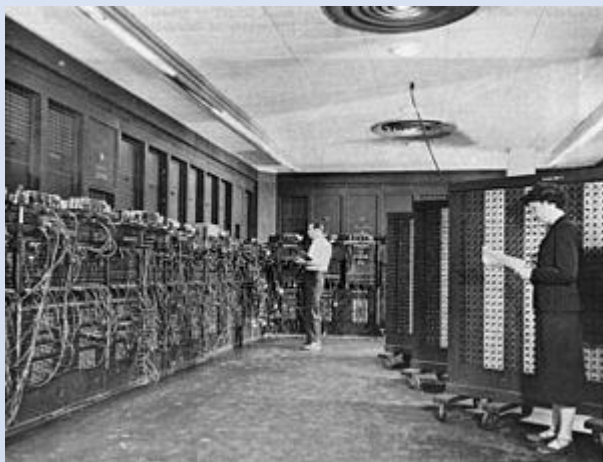
ВВЕДЕННЯ. Історія ОТ

Конрад Цузе (1910- 1995)

Z1 (1938), Z2, Z3



ВВЕДЕННЯ. Історія ОТ



Джон фон Нейман (1903-1957

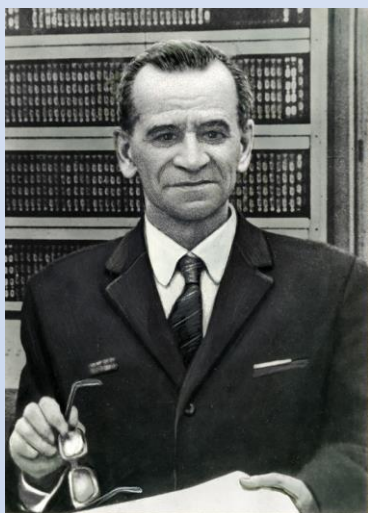
Говард Ейкен (Harvard Mark I, II, III)

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

16

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

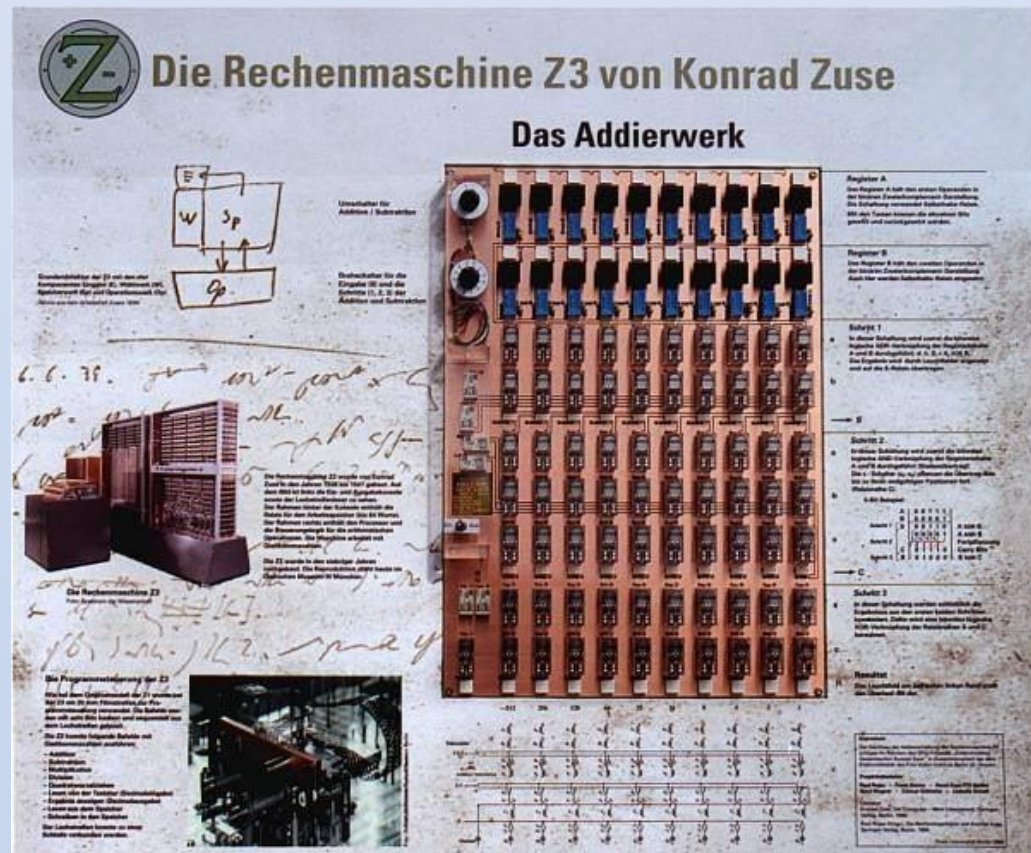
With notes upon the Memoir by the Translator
ADA AUGUSTA, COUNTESS OF LOVELACE

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
     10 b = φ cos (sh)                         (23 m)
     12 fs = φ exp (-a/b)                     (23 m)
      repeat                                     4
      4 f180 = 0                                4
      4 y = 0                                    4
      2 r = 1(1)90                              1
      4 newline                                (90 m)
      7 print (r)2,0                             (150 m)
      2 space                                   (60 m)
     11 s = 2r - 1                               19
     10 f = f(s-1) + 4fs + f(s+1)             22
      9 y = y + hf/3                             (5 m)
      7 print (y) 1,6                           (333 m)
      4 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

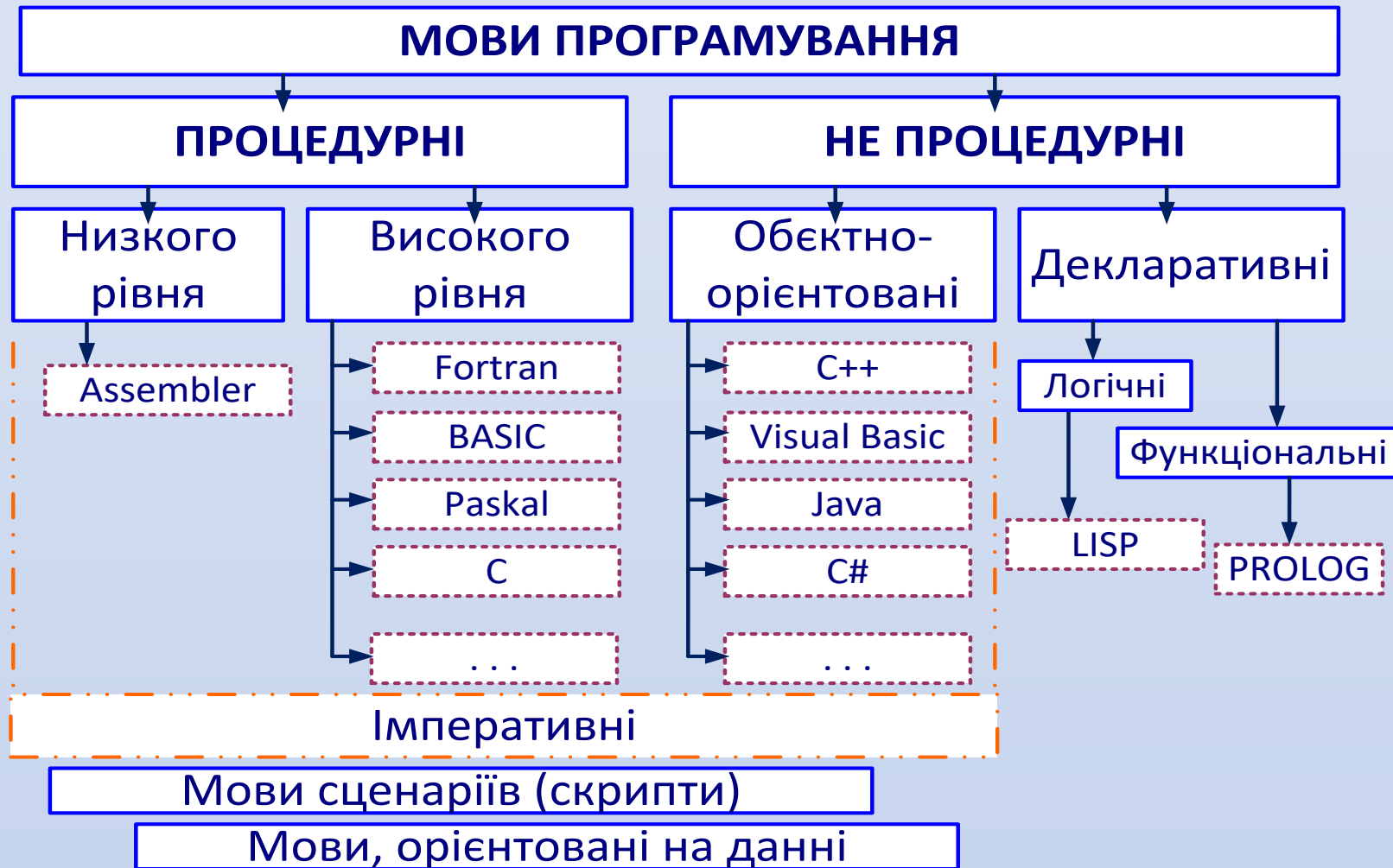
Історія програмування дивись:

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

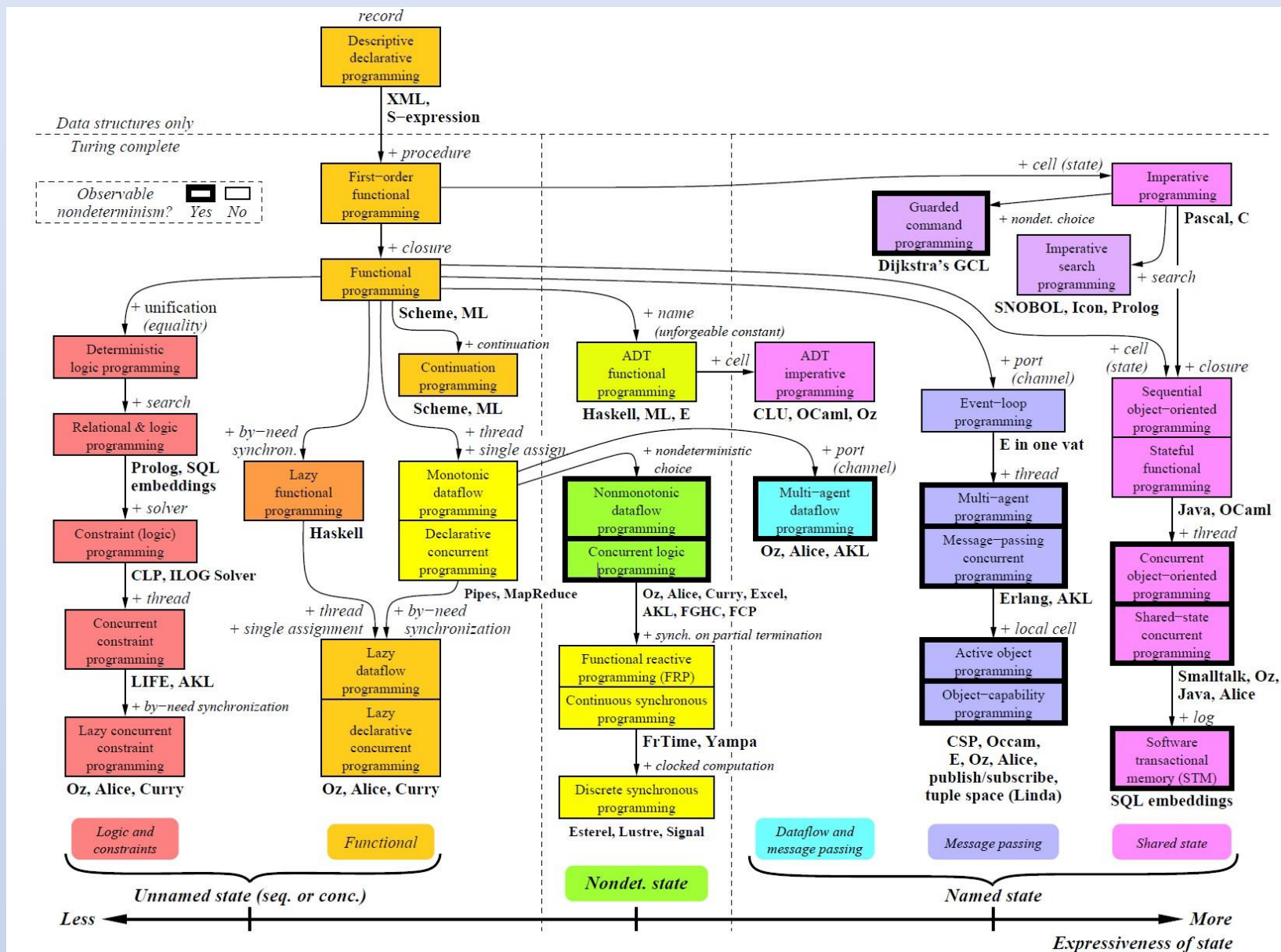
Об'єктно - орієнтовні мови –

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>

ВВЕДЕННЯ. Python

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/>

ВВЕДЕННЯ. Python

Пайтон Дзен

- Гарне краще за потворне. Явне краще за неявне.
- Просте краще за складне. Складне краще за заплутане.
- Плоске краще за вкладене. Розріджене краще за щільне.
- Легкість читання має значення. Особливі випадки не є настільки особливими, щоб порушувати правила. Хоча практичність є важливішою за бездоганність.
- Помилки ніколи не повинні проходити непомітно. Якщо їх приховування не прописано явно.
- Зустрівши неоднозначність, опирайтесь спокусі вгадати.
- Має бути один — і, бажано, *тільки* один — очевидний спосіб зробити це. Хоча спочатку він може бути й не очевидним, якщо ви не голландець.
- Зараз — краще, ніж ніколи. Хоча ніколи, найчастіше, — краще, ніж *просто* зараз.
- Якщо реалізацію важко пояснити — задум поганий. Якщо реалізацію легко пояснити — *можливо*, задум добрий.
- Простори імен — чудова річ, тож робімо їх більше!

ВВЕДЕННЯ. Python



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/>

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.
- **Бейдер Д. Чистый Python. Тонкости программирования для профи.-СПб.: Питер. 2018.-288 с.: ил.**

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.**

The END
Mod 1. Lec 1.