



# ОГЛЯД ГАЛУЗІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Лекція 01  
Основи інформатики, основи програмування та  
алгоритмічні мови



# План лекції

---

- Підхід до задач, які вимагають програмування
- Навчання та професії в галузі IT
- Екосистема розробки програмного забезпечення

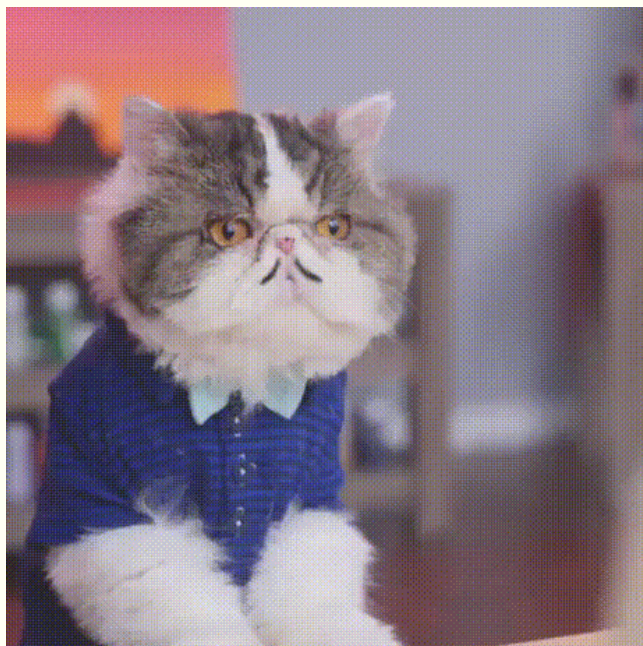


# ПІДХІД ДО ЗАДАЧ, ЯКІ ВИМАГАЮТЬ ПРОГРАМУВАННЯ

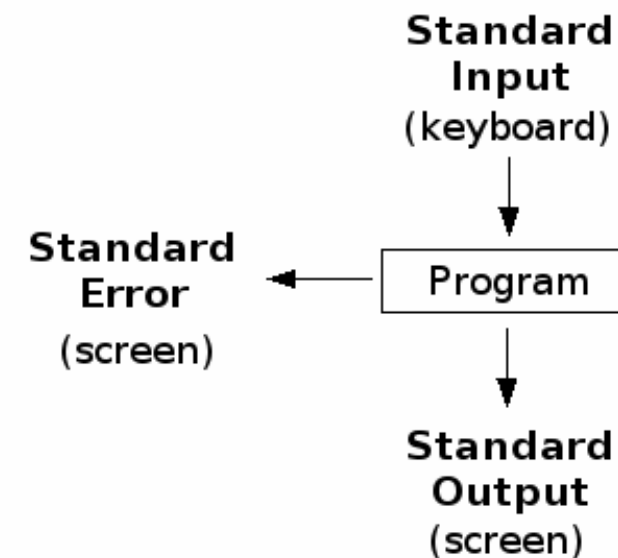
Питання 1.1.

# Суть програмування

- Основна задача – навчити комп'ютер розв'язувати певні задачі, які для людини будуть складними або монотонними.
  - *Computer programming (programming, coding)* – це процес написання, тестування, налагодження та підтримки первинного коду комп'ютерних програм.
  - *Програма* – послідовність інструкцій, які вказують комп'ютеру, як проводити обчислення.
  - Інструкції пишуться спеціальною мовою зі своїми правилами – *мовою програмування*.

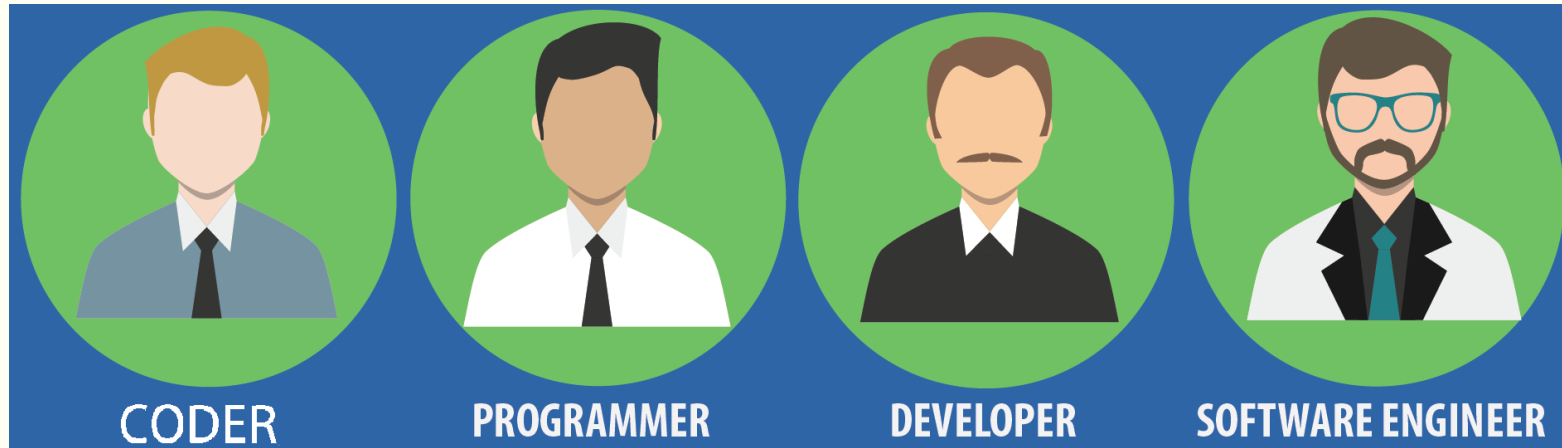


```
17 string sInput;  
18 int iLength, iN;  
19 double dblTemp;  
20 bool again = true;  
21  
22 while (again) {  
23     iN = -1;  
24     again = false;  
25     getline(cin, sInput);  
26     system("cls");  
27     stringstream(sInput) >> dblTemp;  
28     iLength = sInput.length();  
29     if (iLength < 4) {  
30         again = true;  
31         continue;  
32     } else if (sInput[iLength - 3] != '.') {  
33         again = true;  
34         continue;  
35     } while (++iN < iLength) {  
36         if (isdigit(sInput[iN])) {  
37             continue;  
38         } else if (iN == (iLength - 3)) {  
39             continue;  
40         }  
41     }  
42     // ...  
43 }
```



# Люди, які пишуть програмний код

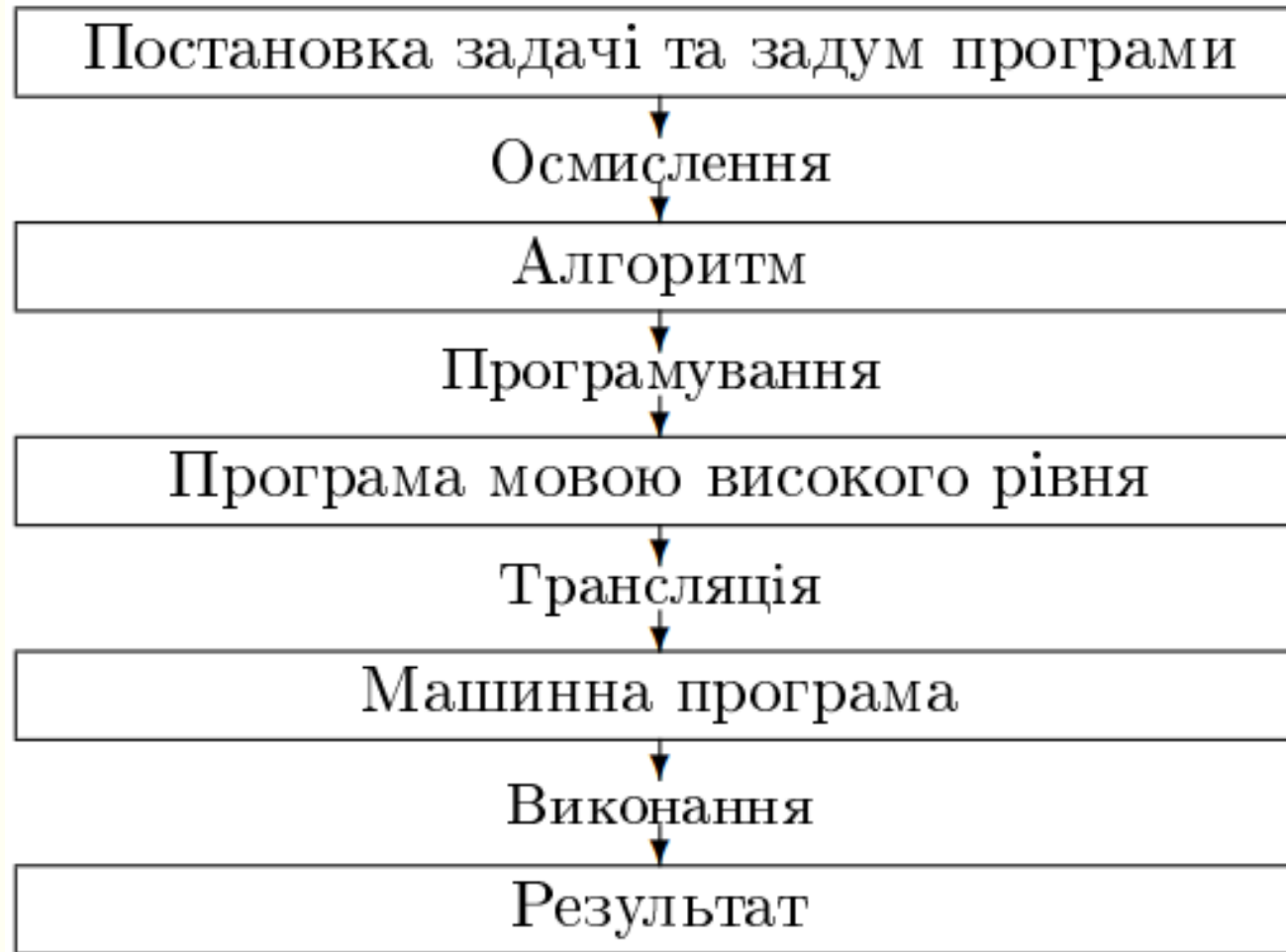
---



- **Кодер (coder)** – пише програмний код, не турбуючись про його красу чи якість.
- **Програміст (programmer)** – пише програми за допомогою мов програмування та розуміє алгоритми їх роботи.
- **Розробник програмного забезпечення (software developer)** – творча особа з широким діапазоном технічних навичок.
  - Написання коду – лише частина його роботи/відповідальності.
  - Додаються обов'язки з документування, концептуального аналізу, інколи проектування, тестування, налагодження та оптимізації коду.
- **Інженер програмного забезпечення (software engineer)** – особа, яка застосовує принципи інженерії програмного забезпечення для проектування, розробки, підтримки, тестування та оцінки програмного забезпечення.

# Процес розв'язку задачі, яка вимагає програмування

---





# Крок 1. Зрозуміти задачу та проаналізувати її

- Уявіть, що у Вас є схожі монети, на яких рік записано римськими цифрами

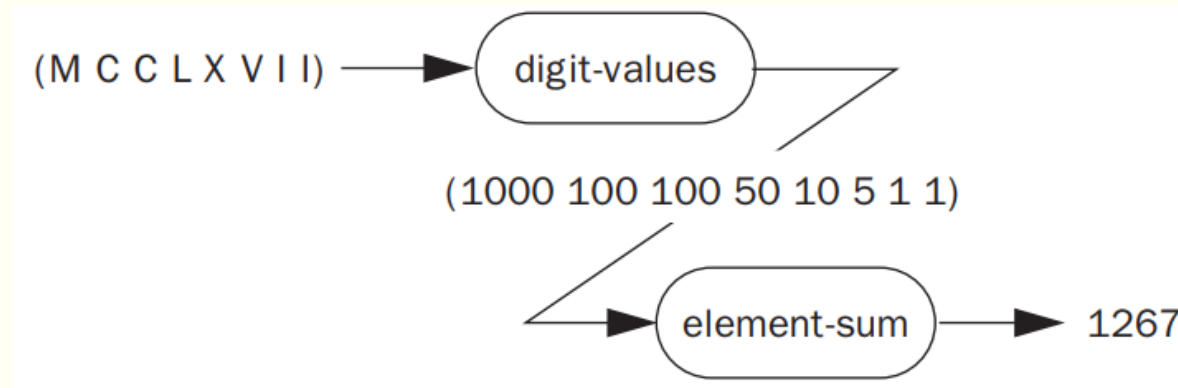


- Хочеться написати програму, яка перетворюватиме римські цифри у зрозумілі десяткові числа
  - Якого року ця монета?
  - Значення, менші за 3999, записуються в римській системі числення за стандартними правилами

Римський запис числа	Десятковий запис числа
I	1
IV	4
V	5
IX	9
X	10
XL	40
L	50
XC	90
C	100
CD	400
D	500
CM	900
M	1000

# Крок 1. Зрозуміти задачу та проаналізувати її

- Правила римського запису чисел:
  - **Порядок цифр:** незростаючий, значення цифр додаються з метою представлення числа, крім префіксів;
  - **Зустрічність цифр:** поспіль можуть стояти максимум 3 однакові цифри (М, С, Х або І), а D, L або V можуть зустрічатись в числі максимум 1 раз;
  - **Префікси:** цифри С, Х або І, які стоять попереду відповідної цифри та віднімаються за такими правилами:
    - С може бути префіксом для М або D, розміщується після попередніх М у числі (ММСМ). Після розрядів М або D можуть бути лише числа до 99;
    - Один Х може бути префіксом для С або L (СССХС, XL). Після розрядів С або L можуть бути лише числа до 9;
    - Один І може бути префіксом для Х або V (ІХ, ІV). Він з'являється наприкінці числа.
- **Уточнене завдання:** маючи на вхід список з розрядів римського числа (М, D, С, С, V, І, І), слід отримати його десяткове представлення – 1707.

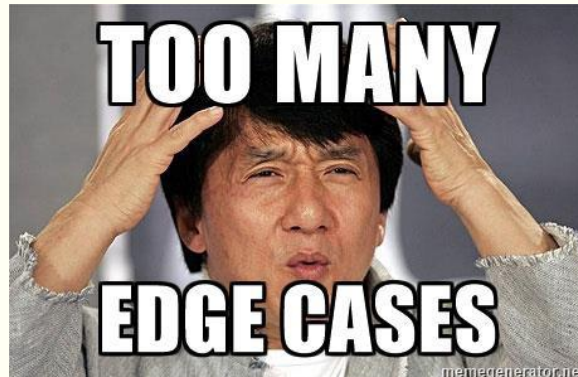




## Крок 2: ретельно розібратись з простими вхідними даними та прикладами

---

- Перевірте своє розуміння:
  - Яке десяткове значення наступних римських чисел: MCMLII, MMCDV, CXCIX?
  - Як буде представлено десяткові числа 1988, 1000 та 1349 за допомогою римських цифр?
  - Визначте, чи коректні такі записи чисел: XMICVC, XLIX, IIX, XIXIV?
- Поради:
  - Спробуйте дуже прості приклади для знаходження результату майбутнього виводу
  - Візьміть більші та складніші вхідні дані, щоб визначити, скільки можливих варіантів ходу виконання програми (use cases) бажано перевірити
  - Продумайте реакцію майбутньої програми на відсутність введених даних
  - Продумайте реакцію майбутньої програми на некоректно введені дані (символи, які не є римськими цифрами, неправильний порядок цифр тощо)
  - Проте майте міру!



## Крок 3: розбийте задачу на менші підзадачі

---

- **Підзадача 1:** перевірити, чи список римських цифр незростаючий (інакше – помилка).  
На вхід – список цифр;
- **Підзадача 2:** перевірити наявність префіксів.
  - Наприклад, для MCMXCIX:
- **Підзадача 3:** організувати віднімання сусідніх цифр для врахування префіксів. На вхід – список цифр;
- **Підзадача 4:** додати проміжні значення та отримати загальну суму.

$$\begin{array}{ccccccc} (1000 & 100 & 1000 & 10 & 100 & 1 & 10) \\ & \underline{\hspace{1cm}} & & \underline{\hspace{1cm}} & & \underline{\hspace{1cm}} & \\ & 900 & & 90 & & 9 & \end{array}$$
  
$$\begin{array}{ccccccc} (100 & 100 & 100 & 10 & 10 & 10 & 1 & 10) \\ & & & & & & \underline{\hspace{1cm}} & \\ & & & & & & 9 & \end{array}$$

## Крок 4: почати програмування

---



- Тримайте в пам'яті:
  - З чого починається програма
  - Що саме зараз Вами розробляється
  - Яка мета цієї розробки
- Рекомендації:
  - Проговоріть для себе метод розв'язку поточної підзадачі
  - Знайдіть найскладніші для реалізації підзадачі та на перших порах ігноруйте їх, розв'язуючи простіші частини. Так трохи зекономиться час на підготовку інфраструктури для реалізації складних програмних частин
  - Зверніть увагу, що в процесі розв'язку задачі можна придумати кращий розв'язок

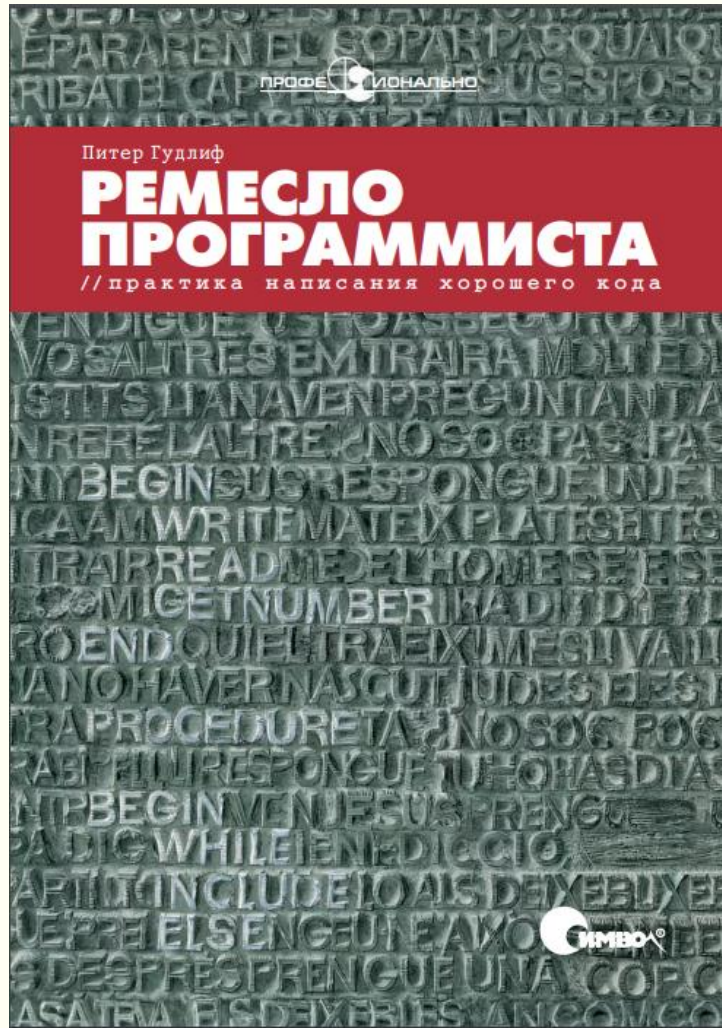
## Крок 5: переглянути готовий код та пошукати можливості з його покращення

---

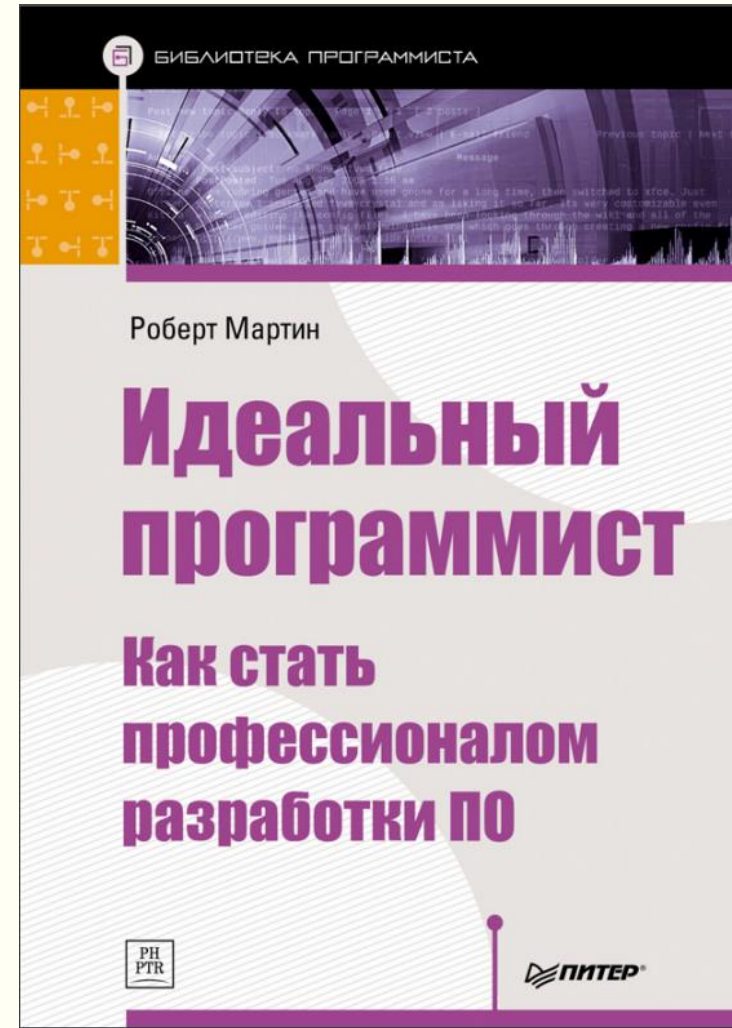
- Спитайте себе:
  - Чи працює цей код для будь-яких можливих вхідних даних?
  - Чи є інший спосіб розв'язати задачу?
  - Чи можна зробити код більш ефективним?
  - Чи зручно читати цей код?
  - Чи зрозуміли б Ви цей код, якби хтось його Вам показав?
  - Чи можна прискорити роботу програми?

# Розумні книги для формування програміста (клікабельні)

---



01.09.2020



@Марченко С.В., ЧДБК, 2020

13





# ДЯКУЮ ЗА УВАГУ!

Наступне запитання: Навчання та професії в галузі ІТ