МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «Комп’ютерне моделювання процесів та систем»

Ефективність та якість архітектурних рішень

**Звіт  
Лабораторна робота №1**

Довгий метод

Виконав ст.гр. ІКМ-М223д

Ковальов Д.Д.

Перевірив викладач

Хорошун А.С.

Харків 2024

Завдання

Тема: Довгий метод.

Мета: Аналіз та оптимізація довгих методів.

Завдання:

Опис проблеми

Виберіть або створіть приклад програмного коду, де існує "довгий метод" – метод, який містить занадто багато коду та може мати неефективну реалізацію.

Аналіз проблеми:

Проведіть аналіз вибраного методу.

Визначте ділянки коду які можно поліпшити.

Оптимізація методу:

Запропонуйте рефакторінг для вибраного методу з метою його покращення.

Використовуйте техніки рефакторінгу надані в теоретичній частині.

Результати:

Проведіть тестування оновленого методу та зіставте результати з оригінальним.

Оцініть отримані покращення та їх вплив на архітектуру системи.

image\_path = '/content/data/hearts 3.png'

img=cv2.imread(image\_path)

img = cv2.resize(img, (img.shape[1] // 2, img.shape[0] // 2))

cv2\_imshow(img)

b, g, r = cv2.split(img)

cv2\_imshow(g)

g=cv2.bitwise\_not(g)

cv2\_imshow(g)

Приклад коду, що розбиває вхідне зображення на RGB канали, та проводить маніпуляції з обраним каналом.

Результат після рефакторингу

def get\_img(image\_path):

  img=cv2.imread(image\_path)

  return img

def BGRsplit(img):

  b, g, r = cv2.split(img)

  return g

def reverse(img):

  return(cv2.bitwise\_not(img))

def main():

  img=get\_img('/content/data/hearts 2.png')

  g=BGRsplit(img)

  result=reverse(g)

Для усіх кроків було виконано метод відокремлення методу. Це дало нам змогу не створювати зайвих змінних, а замість них використати описані методи, то є метод заміни змінної викликом функції.

Даний рефакторинг суттєво покращує розуміння коду та наводить порядок у ньому, ми розуміємо, яка функція що виконує, є розмеження між головною та побічними функціями