МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум №1**

з дисципліни «ВЕБ-технології та ВЕБ-дизайн-2. Веб-орієнтована розробка програмного забезпечення»

на тему: «Вибір підходящих технологій для веб-розробки»

Варіант №8

**Виконав:**

студент гр. БС-83

Кучинський В.В.

**Перевірив:**

ас. Матвійчук О. В.

ас. Давидько О. Б.,

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2020

**Прикріпіть протокол з відповіддю на наступнізапитання:**

1) Виберіть одну з мовпрограмування (C#, Rust, Go, Python) для виконанняпрактикумів, обґрунтуйтесвійвибір.

ВибираюPython, боцевисокорівневамовапрограмування, якуможназастосовувати і для створенняприкладнихпрограм, і для WEB розробки. Продуктивністьплатформи вельми висока, код відрізняється простою і читабельністю.

Інодійогопорівнюють з такими популярними платформами як Ruby, але на відмінувіднього, Pythonвимагаєменшеоперативноїпам'яті,швидшевзаємодіє з процесором.

Pythonвідноситься до найбільшзатребуваних і популярнихМП, про щосвідчатьчисленні рейтинги і аналізпропозицій на ринку розробкипрограмнихпродуктів. Віндоситьпростий, а тому вивченнямови не займе занадтобагато часу

2) Виберітьбібліотеку для роботи з мережею, обґрунтуйтесвійвибір.

Можливо, це буде requests для клієнта и http.server для сервера.Бібліотеки з pythonstandardlibrary, відноснопростий і зрозумілий REST

3) Як в вибраній вами мовіреалізуєтьсяуправлінняпам’яттю, та робота з змінюваними(Mutable) об’єктами/структурами.

В Pythonпам'ятьуправляється автоматично.

• Пам'ять для об'єктів, яківже не потрібнізвільняєтьсяскладальникомсміття.

• Для невеликих об'єктів (<512 байт) Pythonвиділяє і звільняєпам'ять блоками (в блоціможе бути кількаоб'єктів). Чому: операції з блоками пам'яттю через ОС доситьдовгі, а дрібнихоб'єктівзазвичайбагато, і, таким чином, системнівикликивідбуваються не так часто.

• Є два алгоритму складаннясміття: підрахунокпосилань (referencecounting) і збирач на основіпоколінь (generationalgarbagecollector - gc)

• Алгоритм підрахункупосиланьдужепростий і ефективний, але у нього є один великий недолік (крімбагатьохдрібних). Він не вмієвизначатициклічніпосилання

• Зміннізберігаютьпосилання на об'єкти в пам'яті, всерединіоб'єктзберігаєчислове поле - кількістьпосилань на нього (кільказміннихможутьпосилатися на один об'єкт)

• Кількістьпосиланьзбільшується при присвоєнні, передачіаргументів на функцію, вставціоб'єкта в список і т.п.

• Якщо число посиланьдосягло 0, то об'єктвідразувіддаляється (це плюс).

• Якщо при видаленніоб'єктмістивпосилання на іншіоб'єкти, то і тіможутьпіти, якщоцебулиостанніпосилання.

• Змінні, оголошені поза функцій, класів, блоків - глобальні.

• Глобальнізмінніживуть до кінцяпроцесуPython, лічильникїхпосиланьніколи не падає до нуля.

• При виході з блоку коду, посиланнястворенілокальнимизміннимиобластівидимостіцього блоку - знищуються.

В Pythonзмінніоб'єктипередаються в функцію за посиланням. Цеозначає, що не створюєтьсякопіяоб'єкта, а змінною-параметру присвоюєтьсяпосилання на вжеіснуючийоб'єкт. У підсумку, якщо в тіліфункціїоб'єктзмінюється, то цізмінистосуються глобального об'єкта.

4) Як вибрана вами мовавиконуєпаралельніобчислення, як даніобчисленнясинхронізуються.

GlobalInterpreterLock не дозволяє в одному інтерпретаторіPythonефективновикористовуватибільше одного потоку. Захисники GIL стверджують, щооднопоточніпрограми при наявності GIL працюютьнабагатоефективніше. Але наявність GIL означає, щопаралельніобчислення з використаннямбезлічіпотоків і загальноїпам'ятінеможливі. А цедоситьсильнеобмеження в сучасномубагатоядерномусвіті.

5) Пояснітьвідмінністьміжсинхронним та асинхроннимвиконаннямпрограмного коду.

Синхронна програмна модель - цепрограмна модель, коли потоку призначається одна задача і починаєтьсявиконання. Коли завершено виконаннязавданнятодіз'являєтьсяможливістьзайнятисяіншимзавданням. У ціймоделінеможливозупинятивиконаннязавданнящоб в проміжкувиконатиіншезавдання.

Асинхронна модель програмування - на відмінувідсинхронноїпрограмноїмоделі, тут потік одного разу почавши виконаннязавданняможепризупинитивиконаннязберігшипоточний стан і міжтимпочативиконанняіншоїзадачі.

6) Чиможеоднопоточнапрограма бути асинхронною, та багатопоточна синхронною, пояснітьчому.

Може, і такийрезультат є можливим для виконання комп’ютером .

пункт 3 1) коли будуть видалені об'єкти з retain cycle зв'язком? 2) що таке Capasity в розрізі вставки нового елементу в список? 3) які типи в Python є мутабельними? пункт 4 1) як в Python добитись реального паралелелізму? 2) в чому різниця між потоками та процесами? 3) в чому різниця між асинхронним виконанням та concurency? пункт 5 1) що таке корутини? пункт 6 ps: можно описати це хоча б 3-5 реченнями?

пункт 3: 1) При зміненні делегату на слабий (weak), або при завершенні роботи програми 2) Вказівка початкової ємності списку (capacity) дозволяє в майбутньому збільшити продуктивність і зменшити витрати на виділення пам'яті при додаванні елементів 3) До мутабельних (mutable) типів даних відносяться: списки (list), множини (set), словники (dict). пункт 4: 1) необхідно використовувати модуль multiprocessing 2) відмінність полягає в тому, що потоки (одного і того ж процесу) виконуються в загальному просторі пам'яті, а процеси-в окремих просторах пам'яті. 3) Асинхронні виклики використовуються для запобігання "blocking" всередині програми. Такий виклик буде спін-оффом в уже існуючому потоці (наприклад, потік вводу-виводу) і виконувати своє завдання, коли це можливо. У паралельному програмуванні ви як і раніше розбиваєте роботу або завдання, але основні відмінності полягають в тому, що ви запускаєте нові потоки для кожного фрагмента роботи пункт 5: 1) Генераторна функція породжує значення для викликаючої сторони через оператор yield. Однак, генераторна функція також може і отримувати значення від викликаючої сторони. Генератори, які отримують дані від викликаючої сторони називаються корутинами. Всі корутини є генераторами. Але не навпаки. пункт 6: Існують приклади однопоточних асинхронних програм, навприклад: ...do something ...send some async request while (not done) ...do something else ...do async check for results багатопоточність - в цьому сценарії, ми використовуємо багато потоків, які можуть брати завдання і приступати до роботи з ними. У нас є пули потоків (нові потоки також створюються, грунтуючись на потреби та доступності ресурсів) і безліч завдань