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 №13

**Виконав:**

студент гр. БС-83

Розмариця О.А.

**Перевірив:**

ас. Матвійчук О. В.

ас. Давидько О. Б.,

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

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

(підпис викладача)

Київ-2020

**Прикріпіть протокол з відповіддю на наступні запитання:**

***1) Виберіть одну з мов програмування (C#, Rust, Go, Python) для виконання практикумів, обґрунтуйте свій вибір.***

Я обираю Python, оскільки із великої кількості мов програмування з якими мені в тій чи іншій мірі доводилось працювати, саме Пайтон мені більше всіх подобається. Простотою, можливостями, кількістю різних фрейморків, кількістю готових рішень та поширеністю в інтернеті. Рішення більшості проблем у процесі розробки можна знайти в інтернеті на цій мові.

Також вагому роль грає затребуваність на ринку. Python широко використовують у веб-програмуванні, в роботах з БД та machine learning. Хороше знання цієї мови дає можливість обрати для себе напрямок з широкого спектру спеціальностей.

***2) Виберіть бібліотеку для роботи з мережею, обґрунтуйте свій вибір.***

Я обираю стандартну бібліотеку requests, так як вона відносно проста, по ній я знайшов максимально зрозумілі api documentation і в цілому інформації по цій бібліотеці достатньо багато.

Логіка, якою я керуюсь: Python саме в веб напрямку я тільки починаю вивчати, тому я би віддав перевагу максимально простим і поширеним рішенням в цій області, а потім вже, вивчивши і зрозумівши як функціонує взагалі Пайтон в Веб переходив би до більш складних речей.

***3) Як в обраній вами мові реалізується управління пам’яттю, та робота з змінюваними(Mutable) об’єктами/структурами.***

Оскільки Python є високорівневою мовою програмування то користувач не має такого широкого доступу до пам’яті як в тій же С++ чи інших. Пам'ять управляється автоматично в Пайтон.

• Пам'ять для об'єктів, які вже не потрібні звільняється складальником сміття.

• Якщо кількість посилань дорівнює 0 – об'єкт відразу віддаляється.

• Є два алгоритму складання сміття: підрахунок посилань (reference counting) і збирач на основі поколінь (generational garbage collector - gc)

• Алгоритм підрахунку посилань дуже простий і ефективний, але у нього є один великий недолік (крім багатьох дрібних). Він не вміє визначати циклічні посилання

• Для невеликих об'єктів (<512 байт) Python виділяє і звільняє пам'ять блоками (в блоці може бути кілька об'єктів).

• Змінні зберігають посилання на об'єкти в пам'яті, всередині об'єкт зберігає числове поле - кількість посилань на нього (кілька змінних можуть посилатися на один об'єкт)

• Кількість посилань збільшується при присвоєнні, передачі аргументів на функцію, вставці об'єкта в список і т.п.

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

• Глобальні змінні живуть до кінця процесу Python, лічильник їх посилань ніколи не падає до нуля.

• При виході з блоку коду, посилання створені локальними змінними області видимості цього блоку - знищуються.

В Python змінні об'єкти передаються в функцію за посиланням. Це означає, що не створюється копія об'єкта, а змінною-параметру присвоюється посилання на вже існуючий об'єкт. У підсумку, якщо в тілі функції об'єкт змінюється, то ці зміни стосуються глобального об'єкта.

***4) Як вибрана вами мова виконує паралельні обчислення, як дані обчислення синхронізуються.***

В цылому Python є далеко не найшвидшою мовою програмування. Вся справа в його інтерпретаторі – Global Interpreter Lock, який не дозволяє в одному інтерпретаторі Python ефективно використовувати більше одного потоку (насправді є ще деякі обмеження завдяки тому самому інтерпретатору, не пов’язані з роботою пам’яті, але зараз не про них). Проте в інтернеті я знайшов кілька статей, як можна за допомогою модулів collections та multiprocessing обійти цей неприємний недолік і змусити Python виконувати паралельні обчислення.

***5) Поясніть відмінність між синхронним та асинхронним виконанням програмного коду.***

Синхронна модель виконання програмного коду передбачає послідовне виконання задач одна за одною, лише по одній в певний момент часу. Наприклад, якщо ми відправили запит то в синхронній моделі програмування, поки ми не дочекаємось відповіді на наш запит, інші команди не будуть виконуватись.

В асинхронній моделі програмування можливе виконання декількох операцій одночасно. Одна операція може віддаватись на виконання «кудись в інше місце» а головний обробник може йти далі виконуючи інші задачі. Наприклад, обравши на сайті з продажу автомобілів деякі фільтри, користувач може продовжувати взаємодіяти з іншими функціями інтерфейсу, поки «на стороні» відбувається пошук списку автомобілів відповідних до обраних фільтрів.

***6) Чи може однопоточна програма бути асинхронною, та багатопоточна синхронною, поясніть чому.***

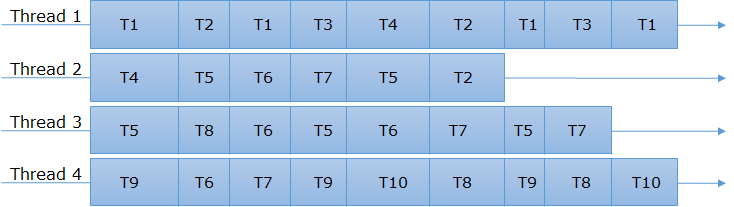
Так, таке можливо. Взагалі виділяють 4 види можливих сценаріїв:

* Синхронний однопоточний
* Синхронний багатопоточний
* Асинхронний однопоточний
* Асинхронний багатопоточний

Для пояснення механізму дій краще підійдуть ілюстрації (поясню різницю асинхронних підходів):



Це *асинхронне* ***однопоточне*** виконання сценарію, при якому ми можемо зупиняти виконання однієї операції і переходити до іншої, не гублячи при цьому результат дії попередньої (синхронний підхід потребує завершення операції після її початку, переривати не можна)



Це *асинхронне* ***багатопоточне***виконання сценарію, при якому одна і та сама операція може виконуватись одночасно в декількох потоках. При цьому потоки можуть мати довільний порядок виконання операцій.