 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Лабораторна робота №1**

з дисципліни «Веб технологій та веб дизайн»

**Виконав:**

студент гр. БС-81

Лукашевич М.Я.

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

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

(підпис викладача)

***Завдання для практикума:***

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

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

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

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

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

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

1)

Python обеспечивает два уровня доступа к сетевым службам. На низком уровне можно получить доступ к базовой поддержке сокетов в базовой операционной системе, которая позволяет реализовывать клиенты и сервера как для протоколов, ориентированных на соединение, так и без установления соединения.

Python также имеет библиотеки, обеспечивающие более высокий уровень доступа к определенным сетевым протоколам уровня приложения, таким как FTP, HTTP и т. д.

2) Django

Высокоуровневый фреймворк, предназначенный для быстрой и эффективной разработки. Его архитектура выстроена так, чтобы максимально экономить ваше время и силы, а в будущем — время и деньги заказчиков. Основные принципы философии Django — DRY и rapid development — стимулируют переиспользование кода и уменьшают избыточность.

3)

В Python диспетчер памяти отвечает за такие задачи, периодически выполняя очистку, выделение и управление памятью. В отличие от C, Java и других языков программирования, Python управляет объектами с помощью подсчета ссылок. Это означает, что диспетчер памяти отслеживает количество ссылок на каждый объект в программе. Когда счетчик ссылок объекта падает до нуля, что означает, что объект больше не используется, сборщик мусора (часть диспетчера памяти) автоматически освобождает память от этого конкретного объекта.

Пользователю не нужно беспокоиться об управлении памятью, поскольку процесс выделения и удаления памяти полностью автоматизирован. Восстановленная память может использоваться другими объектами.

Изменяемые объекты не обладают постоянством. Они скорее напоминают контейнер для хранения: контейнер остаётся на месте, а вот содержимое может сильно измениться. Ну, например, если передать список в «чужую» функцию, то его могут испортить до неузнаваемости.

4)

Python предоставляет множество библиотек и инфраструктур, которые служат для удобства высоко производительных вычислений. Однако, выполнение параллельного программирования при помощи Python может быть достаточно коварным по причине GIL (Global Interpreter Lock, Глобальной блокировке интерпретатора).

На практике, большинство распространённых и наиболее широко применяемых интерпретаторов используют интерпретатор Python, CPython, разработанный на языке программирования C. Для интерпретатора CPython требуется GIL для сохраняющих поток (thread- safe) операций. Такое использование GIL подразумевает что вы подсчитываете глобальные блокировки всякий раз когда пытаетесь выполнить доступ к любому объекту Python, содержащемуся внутри потоков. Причём лишь один поток за раз способен выдать запрос на блокировку некого объекта Python или API C.

К счастью, дела не так плохи, потому что вне пределов этого царства GIL мы свободно можем использовать параллелизм. Эта категория содержит все те темы, которые мы обсудим с своих последующих главах, включая многопроцессность, распределённые вычисления и вычисления GPU.

5)

В синхронном коде каждая операция ожидает окончания предыдущей. Поэтому вся программа может зависнуть, если одна из команд выполняется очень долго. Асинхронный код убирает блокирующую операцию из основного потока программы, так что она продолжает выполняться, но где-то в другом месте, а обработчик может идти дальше.

6)

Синхронный проограмма может быть многопоточным, а асинхронный однопоточным нет.