 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-81

Погребенко В. О.

**Перевірили:**

[ас. Давидько О. Б.](http://rozklad.kpi.ua/Schedules/ViewSchedule.aspx?v=b92c0571-df7d-410d-8013-82552f62f3cf), [ас. Матвійчук О. В.](http://rozklad.kpi.ua/Schedules/ViewSchedule.aspx?v=dab62327-7bea-4b83-ba3f-a2aa2848e036)

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

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

(підпис викладача)

Київ-2020

**Завдання:**

Прикріпіть протокол з відповіддю на наступні запитання:

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.

**Відповіді на запитання:**

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

Було обрано мову програмування Golang(Go), по наступним причинам:

* Наявність певного досвіду роботи з golang у IT компанії.
* За допомогою golang можна просто організовувати паралелізм за допомогою goroutines та channels.
* За допомогою golang можна порівняно просто писати веб-сервіси.
* У golang просто проводити юніт-тестування за допомогою вбудованого фреймворку testing (import "testing").

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

Для роботи з мережею був обраний стандартний пакет net, та написаний на ньому пакет gorilla (<https://github.com/gorilla>):

Причини вибору:

* + Gorilla – один з найстаріших та найбільших toolkit’ів для golang, що має дуже високий рейтинг.
  + Gorilla має найбільше англійське community з усіх інших подібних пакетів.
  + Gorilla має багато пакетів, серед яких є пакети для роботи з маршрутизацією та websocket’ами, що будуть використовуватися у лабораторних.
  + Gorilla сумісна з базовим пакетом net та використовується разом з ним.
  + net - net/http – стандартний пакет, що має все необхідне для того, щоб виконувати http запити.

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

**Як виділяється пам’ять:**

Пам’ять виділяється в go під час використання функцій new and make, оголошення змінних, зміни розмірів slice’у, etc.

**Де** **виділяється пам’ять:**

Golang сам керує тим, де і як виділити пам’ять, і компілятор golang зазвичай намагається виділяти пам’ять у stack’у. Але, наприклад, коли змінна занадто велика, чи на неї робиться посилання, або якщо вона передається до функції, то тоді для цієї змінної буде виділена пам’ять у heap’і (value will escape to the heap).

Крім того, в офіційній специфікації golang сказано, що: «місце зберігання, вибране реалізацією, не має значення для семантики мови.».

**Робота з змінюваними(Mutable) об’єктами/структурами:**

*Створення:*

Slice’и, map’и можна створювати за допомогою make:

m = make(map[string]string)

s = make([]string, size)

Об’єкти можна створювати за допомогою new, structName{}, чи var:

example := new(ExampleStruct)

example := &ExampleStruct{}

var example ExampleStruct

example := ExampleStruct{}

*Додавання до Slice’у чи map’у:*

var exampleSlice

exampleSlice = append(exampleSlice, “test”)

exampleMap = make(map[string]string)

exampleMap[“test”] = “test”

*Видалення з Slice’у чи map’у*:

delete(exampleMap, “key”)

slice := append(slice[:s], slice[s+1:]...)

*Оголошення структури і зміна полів об’єкту:*

*type* ExampleStruct struct {

test string

}

var example ExampleStruct

example.test = “”

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

Паралельні обчислення реалізуються за допомогою Goroutine:

*func hello() {*

*fmt.Println("Hello world")*

*}*

*func main() {*

*go hello()*

*}*

Синхронізація зазвичай відбувається за допомогою:

* channels – канали використовуються для обміну інформацією між горутинами та для іх синхронізації за допомогою операцій запису та читання з каналу (ch <- 1, someVar <-ch)
* sync – за допомогою пакету sync та його функціоналу. Наприклад: sync.Mutex (m.Lock(), m.Unlock()), sync.WaitGroup(wg.Add(), wg.Wait()), sync.atomic(AddUint…, LoadUint…), etc.

1. ***Поясніть відмінність між синхронним та асинхронним виконанням програмного коду.***

Під час виконання синхронного коду все операції відбуваються послідовно, одна за одною. Під час виконання асинхронного коду певні задачі можуть виконуватися паралельно, не очікуючи одна-одну.

Також, на відміну від синхронного коду, під час виконання асинхронного коду, треба забезпечувати потокобезпечність, щоб уникнути data race’и та race condition’и.

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

Однопоточна програма може бути асинхронною, це називається «Concurrency**».** У такому випадку задачі виконуються в одному потоці паралельно, але не одночасно.

Так, багатопоточну програму можна синхронізувати у певний момент виконання програми (об’єднати треди в 1), можна використовувати мьютекси для забезпечення того, щоб певний блок коду не виконувався більше ніж одним потоком одночасно, або взагалі запустити програму з 1 потоком виконання якщо це можливо (example.exe –threads=1).