



Силабус дисципліни «Системне програмування»

Викладач: Бурлаченко Іван Сергійович
старший викладач кафедри комп'ютерної інженерії
факультету комп'ютерних наук ЧНУ імені Петра Могили.

Очікувані результати навчання

В результаті вивчення дисципліни студент

має знати:

- об'єктно-орієнтовану модель мови Go і навчитися проектувати згідно з нею програмні компоненти;
- основні складності розробки конкурентного програмного коду і способи їх усунення;
- як організувати декомпозицію коду проєкту на складові компоненти для зручності подальшої підтримки та повторного використання;
- як вирішувати системні завдання на рівні, достатньому для промислового застосування;

має вміти:

- впроваджувати програмні залежності у конфігурацію web-проєктів;
- спроектувати модель бази даних у web-застосунку на Go і налаштувати пул з'єднань;
- кешувати ваші шаблони, відображати динамічні дані, створювати призначені для користувача функції і обробляти помилки під час виконання;
- використовувати і налаштовувати сеанси для збереження даних між запитами;
- налаштувати сервер HTTPS і налаштувати його для підвищення продуктивності і безпеки;
- впровадити аутентифікацію і авторизацію, та запобігти SQL

Обсяг: буде визначено після здійснення процедури вибору студентами вибіркових дисциплін.

Мета: навчитися використовувати системний підхід до проектування архітектури великих програмних продуктів, працювати в командах back-end розробників.

Оригінальність навчальної дисципліни

Матеріали дисципліни представляють сукупний досвід інженерії програмного забезпечення багатьох tech-лідерів проєктів та solution-архітекторів міжнародних ІТ компаній.

Зміст дисципліни

Тема 1. Environment setup. Як встановити та налаштувати компілятор мови Go.

Приклад написання, компіляції та запуску простої програми на мові Go.

Тема 2. Go IDEs. Які інтегровані середовища розробки можна використовувати для написання коду на мові Go, як їх встановити і налаштувати.

Тема 3. Go tools. Основний інструментарій підтримки процесу розробки. Як скомпілювати і запустити програму на мові Go, як отримати доступ до документації, а також надати документацію в власний код.

Тема 4. Directory structure. Go modules. Packages. Як організувати структуру директорії проєкту так, щоб вона відповідала загальноприйнятим правилам.

Способи декомпозиції коду на окремі складові для зручності подальшої підтримки та повторного використання.

Тема 5. Installing dependencies. Використання готових рішень замість написання власного коду. Управління залежностями і розподілене зберігання пакетів.

Тема 6. Data types and variables declaration. Zero values. Type conversions

Подання примітивних значень в мові і виконання базових операцій над ними.

Тема 7. Standard library: fmt, log, strings, time, sort, http, math, regexp

Вбудовані механізми мови для вирішення повсякденних завдань.

ін'єкціям, CSRF, XSS, перехопленню кліків.

Тема 9. Containers: arrays, slices, maps. Як зберігати послідовності значень: масиви, зрізи і хеш-кодування, їх внутрішнє уявлення і приклади використання.

Тема 10. Branching and loops. break, continue, fallthrough, goto
Опис алгоритмів за допомогою циклів і розгалужень, відміну синтаксичних конструкцій в Go від інших мов програмування.

Тема 11. Functions. Multiple return values. Variadic parameters.
Function literals and closures

Визначення функцій і передача параметрів. Функції з будь-якою кількістю аргументів. Функціональні можливості мови, анонімні функції і замикання.

Тема 12. Types declarations. Constants, Iota. Structures. Embedded structs. Anonymous structs. Nested structs

Способи визначення призначених для користувача типів і констант, опис послідовностей значень. Як описати сутності предметної області за допомогою конструкцій мови.

Тема 13. Pointers. Value and references types

Управління пам'яттю і покажчики.

Тема 14. Defer statement Оператор відкладеного виклику функцій.
Error handling. Panic, recover Як виявити і обробити помилки часу виконання. Способи написання надійного коду.

Тема 15. Methods and interfaces. Method expressions

Поняття інтерфейсу і його реалізації. Побудова об'єктно-орієнтованого коду на основі інтерфейсів. Методи.

Тема 16. Concurrency vs parallelism. Goroutines. Memory model
Як робити кілька речей одночасно. Організація конкурентного виконання на основі горутін. Модель пам'яті в Go.

Тема 17. Channels. Blocking and unblocking channels. select statement. Спосіб написання надійного конкурентного коду: обмін даними між горутінами на основі каналів.

Тема 18. Waitgroups, mutexes, atomics Синхронізація окремих частин програми.

Тема 19. Reading and writing files Як зберігати дані між окремими запусками програми.

Тема 8. Testing and benchmarking. Як гарантувати стабільність програмного коду за допомогою модульних тестів, а також знайти найбільш ефективне рішення задачі.

Тема 20. JSON encoding and decoding. Обмін даними з програмами, написаними на інших мовах.

Тема 21. Databases. PostgreSQL. MongoDB. Зберігання даних в реляційних базах.

Тема 22. REST application example. Приклад повноцінного back-end застосунку. Як створити сучасну структуру маршрутизації запитів, яка дотримується принципів REST.

Тема 23. Unsafe package. Розширення мови для виконання небезпечних функцій.

Тема 24. Основи Web. Запуск серверу, створення обробників, відправлення відповіді, направляти запити і обслуговувати статичні файли.

Тема 25. Як створити ідіоматичну і масштабовану структуру для вашого веб-застосунку.

Тема 26. Як кешувати HTML-шаблони, відобразити динамічні дані, створювати призначені для користувача функції і обробляти помилки під час виконання.

Управління сесіями.

Тема 27. Використання HTTPS. Як правильно налаштувати сервер HTTPS і налаштувати його для підвищення продуктивності і безпеки.

Тема 28. Запобігання поширених вразливостей - Як запобігти впровадження SQL ін'єкцій, CSRF, XSS, перехоплення кліків і т.д.

Тема 29. Аутентифікація і авторизація - Як безпечно шифрувати паролі користувачів і додавати функції реєстрації, входу і виходу.

Тема 30. Request context - Як використовувати контекст Go.Context для передачі даних між проміжним програмним забезпеченням і обробниками.

Пререквізити

Програмування, об'єктно-орієнтоване програмування, алгоритми і структури даних

Постреквізити

інженерія програмного забезпечення, крос-платформне програмування, web-технології, технології розробки мобільних застосунків, паралельне програмування, методи машинного навчання, формальні граматики та компілятори, розпізнавання образів, системи та методи прийняття рішень.

Семестровий контроль: залік/іспит

Оцінювання:

За семестр: 70/40 балів

За залік/іспит: 30/40 балів

Види робіт:

Лабораторні роботи – 60/50 балів (12x5/10x5 б.)

Індивідуальне проєктне завдання – 10 балів.

Технічне забезпечення

Компілятор мови Go

VS Code IDE

Moodle, Google Meet, Zoom, Webex, профілі соціальних мереж

Політика щодо дедлайнів

Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку.

Політика щодо академічної доброчесності

Передбачає самостійне виконання лабораторних робіт та індивідуального проєктного завдання. Списування під час заліку (в т.ч. із використанням мобільних пристроїв) заборонено. У разі виявлення плагіату або списування роботи не зараховуються.

Критерії оцінювання лабораторних робіт

- вміння самостійно вирішувати поставлені задачі та представляти рішення у вигляді оптимального програмного коду.
- вміння користуватися специфічними інструментальними засобами для створення програмного забезпечення, що задовольняє SOLID принципам.
- вміння лаконічно та логічно формувати відповіді на запитання, пов'язані з виконаними лабораторними роботами.

Критерії оцінювання індивідуального проєктного завдання

- формулювання актуальності, проблеми, мети і завдань, практичного значення;
- вміння самостійно створювати оптимальний програмний код для вирішення поставленої задачі з використанням патернів програмування.;
- вміння користуватися специфічними інструментальними засобами для розробки ПЗ з ефективною архітектурою;
- якість доповіді студента (форма доповіді, зміст, доказова база, висновки) ;
- повнота та логічність відповідей на поставлені питання.