



# Силабус дисципліни «Програмування комп'ютерних мереж»

**Викладач:** Пузирьов Сергій Володимирович  
канд. фіз.-мат. наук, доц. кафедри комп'ютерної інженерії  
факультету комп'ютерних наук ЧНУ імені Петра Могили.

## Очікувані результати навчання

В результаті вивчення дисципліни студент  
*має знати:*

- модель OSI;
- стек протоколів TCP/IP;
- архітектуру мережі;
- архітектуру мережевих додатків;
- мережевий та апаратний порядок байт;
- серіалізація даних;
- засоби POSIX/WinSock2 для роботи з IP-адресами;
- засоби POSIX/WinSock2 для створення TCP/UDP-серверів;
- засоби POSIX/WinSock2 для створення TCP/UDP-клієнтів;
- засоби POSIX/Windows для багатопоточного програмування;
- засоби POSIX/Windows для роботи з протоколами ICMP/IGMP;
- архітектуру паралельних TCP/UDP-серверів;
- засоби Java/C# для паралельного програмування;
- засоби Java/C# для роботи з IP-адресами;
- засоби Java/C# для створення TCP/UDP-серверів;
- засоби Java/C# для створення TCP/UDP-клієнтів;
- засоби Java/C# для створення групових мережевих додатків;
- засоби Java/C# для роботи з протоколом HTTP;
- архітектуру мережевих корпоративних додатків на базі Java EE

**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 простий 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.

### **має вміти:**

- аналізувати предметну область, виділяти її сутності;
- проектувати архітектуру мережевих додатків;
- конвертувати дані з мережевого порядку байт у апаратний і навпаки;
- працювати з IP-адресами та DNS;
- створювати одно- та багатопотокові TCP/UDP-сервери;
- створювати одно- та багатопотокові TCP/UDP -клієнти;
- розробляти користувацькі протоколи на основі TCP/IP;
- працювати з віддаленими мережевими ресурсами та зберігати їх на локальному комп'ютері;
- створювати найпростіші мережеві корпоративні додатки.

**Тема 1. Фізична організація мережі.** Модель OSI. Лінії глобальної мережі. Протокол Ethernet. Фізичні компоненти мережі. Топологія мереж. Рівні організації мережі за моделлю OSI. Апаратний та мережевий порядок байт

**Тема 2. Мережеві протоколи та служби.** Базові протоколи. Стек протоколів TCP та UDP. Дейтаграми та поточкові протоколи. Поняття сокету. Протоколи Інтернету. Стек протоколів IPv4 і IPv6. Передача повідомлень у мережі. Способи доступу до мережевих об'єктів.

**Тема 3. Адресація в комп'ютерних мережах.** Розрізнення імен. IPv4 адресація. IPv6 адресація. Порти. Адресна структура sock\_addr. Використання inet\_address. Розрізнення імен. Структура hostent, використання gethostbyname. Створення утиліти nslookup.

**Тема 4. Поточкові сокети TCP.** Конфігурування сокетів. Структура TCP-паketу. Створення поточкового сокету. Закриття сокетів. Обробка помилок. Блокуючий та неблокуючий режими роботи сокетів.

**Тема 5. Дейтаграмні сокети.** UDP-сервер та клієнт. Особливості протоколу UDP. Створення UDP-сокетів. Структура UDP-паketу. Відправлення повідомлень по UDP. Функція sendto. Прийом повідомлень по UDP. Функція recvfrom.

Тема 8. Testing and benchmarking. Як гарантувати стабільність програмного коду за

**Тема 6. Багатопотокові TCP-сервери.** Використання ресурсів. Ідіоми багатопоточного програмування Windows API та POSIX API. Синхронізація доступу до ресурсів - м'ютекси, семафори, події, умовні змінні. Обробка вхідних підключень у окремих потоках. Обмеження кількості підключень та потоків.

**Тема 7. Службові протоколи TCP/IP стеку.** Передача даних unicast, multicast, broadcast. Протокол IGMP. Мережеві групи та сокети. Події у групах. Створення групових чатів. Тестування мережі за протоколом ICMP. Створення утиліти ping.

**Тема 8. Синхронне/асинхронне введення/виведення Java/.Net.** Байтові та символні потоки. Буферизація потоків введення/виведення. Серіалізація об'єктів. Робота з файлами та каталогами. Обробка помилок. Концепція неблокуючого введення/виведення. Канал даних. Вибір даних. Асинхронне зчитування/запис даних. Обробка повідомлень. Виключення.

**Тема 9. Високорівневі абстракції для роботи з URL.** Класи URL, URLConnection у Java (відкриття з'єднання з ресурсом, синхронне та асинхронне зчитування даних з ресурсу). З'єднання по протоколу HTTP/HTTPS. Аутентифікація. Обробка помилок.

**Тема 10. Багатопоточне програмування на платформі Java/.Net.** Абстракція потоку. Інтерфейс Runnable. Створення та завершення потоку. Обмін даними між потоками. Синхронізація доступу. Паралельні колекції. Пул потоків. Виконавці. Асинхронне виконання операцій. Майбутні результати. Виключення у потоках.

**Тема 11. TCP-сервери/клієнти засобами Java/.Net.** Адресація. Клас InetAddress. Розрізнення імен. Класи клієнтських та серверних сокетів. Конфігурування сокетів. Створення багатопоточних та асинхронних клієнт-серверних додатків в Java/.Net.

**Тема 12. Створення мережевих додатків на основі протоколу UDP засобами Java/.Net.** Робота з дейтаграмами в Java/.Net. Створення групових розсилок. Мережевий чат. Обробка помилок.

**Тема 13. Робота з HTTP-протоколом на платформі Java.** Сервлет. Контейнер сервлетів. Життєвий цикл сервлету. Класи Servlet, HttpServlet. Конфігурування сервлету. Отримання даних сервлетом. Дані форми. Розгортання сервлетів.

### **Пререквізити**

Програмування, системне програмування, об'єктно-орієнтоване програмування, алгоритми і структури даних, комп'ютерні мережі

**Семестровий контроль:** залік/іспит

**Оцінювання:**

За семестр: 70/60 балів

За залік/іспит: 30/40 балів

**Види робіт:**

Лабораторні роботи – 30 балів (10x3 б.).

Індивідуальні завдання до лабораторних робіт – 40/30 балів (10x4/10x3 б.)

**Технічне забезпечення**

MinGW/TDM/MSys SDK, JDK >= 8, .Net/.Net Core >= 3.0

IntelliJ Idea >= 2019, Visual Studio >= 2017, VS Code IDE

CMake, Git, GitLab, GitHub

**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. Списування під час заліку (в т. ч. із використанням мобільних пристроїв) заборонено. У разі виявлення плагіату або списування роботи не зараховуються.

**Пореквізити**

Кібербезпека, Інтернет речей, web-технології (Spring)

**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 використовувати сторонні бібліотеки для вирішення типових завдань;
- вміння розміщувати проекти лабораторних робіт у системах контролю версій;
- вміння документувати проекти лабораторних робіт;
- вміння лаконічно та логічно формувати відповіді на запитання, пов'язані з виконаними лабораторними роботами.