

Компетентності рівня Junior для Golang Developer		Компанія
	Загальна оцінка компетентності (max. 3)	SoftServe
1. Основи Golang та підготовка середовища		
1.1 Встановлення та налаштування середовища Знає, як встановлювати Go на різних платформах та налаштовувати середовище розробки. Розуміє, як обрати та налаштувати IDE для ефективної роботи з Go. Має базове уявлення про структуру програм на Go, включаючи основні компоненти, файлову організацію та перший запуск простих програм.	3	3
1.2 Основи синтаксису Go Розуміє основні типи даних у Go, правила оголошення змінних та базові оператори. Знає правила іменування змінних і функцій для забезпечення читабельності коду. Має навички написання простих програм з використанням базового синтаксису.	3	3
1.3 Функції та пакети Вміє оголошувати та використовувати функції у програмах на Go, включаючи передачу аргументів і повернення значень. Розуміє принципи роботи з пакетами: їх імпорт, організацію та використання для розділення логіки програми на модулі.	3	3
1.4 Умовні конструкції Розуміє синтаксис і використання умовних конструкцій if, else та switch у Go. Знає, як застосовувати ці конструкції для побудови умовної логіки, та має навички написання практичних сценаріїв з використанням умов.	3	3
1.5 Розуміння модулів Go Розуміє концепцію модулів у Go: їх створення, управління та імпорт у проекти. Вміє працювати з go.mod для налаштування залежностей проекту та організації коду у вигляді модулів, що спрощують роботу над масштабними програмами.	2	2
2. Робота з даними та структурами		
2.1 Масиви та зрізи Розуміє основи роботи з масивами та зрізами у Go. Вміє створювати та ініціалізувати зрізи, виконувати операції додавання, видалення та модифікації елементів. Має навички оптимального використання зрізів для роботи з динамічними даними.	3	3
2.2 Мапи у Go Розуміє, як оголошувати та ініціалізувати мапи у Go. Вміє працювати з ключами та значеннями: додавання, оновлення, видалення даних, перевірка наявності ключів. Має навички використання мап для зберігання асоціативних даних.	3	3
2.3 Оголошення та ініціалізація даних Знає особливості оголошення змінних, констант, структур у Go. Розуміє різні способи ініціалізації та їх застосування у реальних завданнях. Вміє організовувати дані для зручної та ефективної роботи з ними.	3	3

2.4 Методи для типів даних Розуміє, як створювати методи для структур у Go. Вміє використовувати методи для роботи з даними, пов'язаними зі структурами, та забезпечувати логічну організацію коду.	3	3
2.5 Розуміння інтерфейсів Знає, що таке інтерфейси у Go, їх роль у програмуванні та як їх оголошувати. Вміє використовувати інтерфейси для абстрагування логіки та створення гнучкого коду, який легко розширювати.	2	2
2.6 Поліморфізм через інтерфейси Розуміє, як інтерфейси реалізують поліморфізм у Go. Знає практичні сценарії застосування поліморфізму, зокрема створення функцій, що працюють з різними типами через загальні інтерфейси.	2	2
2.7 Тип any та дженерики Розуміє, як використовувати тип any для роботи з узагальненими даними. Знає нововведення дженериків у Go, їх роль у забезпеченні типобезпеки, а також має навички застосування дженериків для створення гнучкого коду.	2	2
3. Обробка помилок та введення/виведення		
3.1 Обробка помилок у Go Розуміє основи створення, повернення та обробки помилок у Go за допомогою механізму error. Вміє працювати з вбудованими та кастомними помилками, застосовувати перевірку помилок для забезпечення надійності коду.	3	3
3.2 panic та recover Розуміє механізми panic і recover для обробки критичних ситуацій у програмі. Вміє генерувати помилки через panic і безпечно їх обробляти за допомогою recover. Знає, коли доцільно застосовувати ці механізми.	3	3
3.3 Функції введення/виведення Знає основи роботи з введенням та виведенням даних у Go. Вміє читати дані з консолі та виводити результати, забезпечуючи коректну взаємодію з користувачем.	1	1
3.4 Файловий ввід/вивід Розуміє, як працювати з файлами у Go. Вміє читати та записувати дані у файли, використовувати буфери для підвищення ефективності обробки даних.	3	3
3.5 Логування з пакетом log Знає, як використовувати стандартний пакет log для запису повідомлень про події. Вміє налаштовувати кастомні логери, формувати зрозумілі та інформативні журнали для налагодження та моніторингу.	2	2
3.6 Обробка помилок у функціях I/O Розуміє, як правильно обробляти помилки при роботі з ввід/виводом та файлами. Вміє забезпечувати надійність роботи програм шляхом перевірки помилок на кожному етапі операцій з даними.	3	3
4. Структура проекту та тестування		
4.1 Структура проекту в Go Знає основи організації проектів у Go, включаючи створення модулів, директорій та файлів. Розуміє принципи розподілу логіки між пакетами, дотримання стандартів і конвенцій у Go. Вміє структурувати код для забезпечення зрозумілості та зручності в роботі над проектами.	2	2
4.2 Операції з файлами Володіє базовими навичками роботи з файлами у Go, включаючи читання, запис, створення та видалення файлів. Розуміє, як працювати з файловою системою для управління ресурсами.	3	3
4.3 Основи тестування в Go Розуміє основні принципи тестування у Go. Вміє писати юніт-тести за допомогою пакета testing, створювати структури тестів та застосовувати основні підходи для перевірки функціональності коду.	3	3

4.4 Запуск та аналіз тестів Знає, як запускати тести у Go за допомогою стандартних інструментів, перевіряти покриття коду (coverage) та аналізувати результати тестування. Володіє рекомендаціями щодо ефективного тестування та покращення якості коду.	3	3
4.5 Модульне тестування роботи з файлами Вміє писати модульні тести для функцій, які працюють з файлами. Забезпечує перевірку коректності операцій з файлами, тестує обробку помилок та правильність управління ресурсами.	3	3
5. Алгоритми та структури даних		
5.1 Основи структур даних Знає ключові структури даних, такі як масиви, списки, дерева та хеш-таблиці. Розуміє їх призначення, способи застосування та реалізацію в Go. Володіє базовими навичками роботи зі структурами даних у реальних задачах.	3	3
5.2 Алгоритми сортування Вміє реалізовувати базові алгоритми сортування в Go, включаючи Bubble Sort, Quick Sort та Merge Sort. Розуміє принципи їх роботи, випадки використання та відмінності між ними.	3	3
5.3 Часова та просторова складність Розуміє поняття часової та просторової складності алгоритмів. Вміє аналізувати ефективність алгоритмів, оцінювати їх продуктивність та вибирати оптимальні рішення для різних завдань.	3	3
6. Конкурентність та асинхронність		
6.1 Основи конкурентності у Go Розуміє основні концепції конкурентності в Go, включаючи поняття горутин. Знає, як використовувати горутини для створення багатопотокових додатків, і розуміє переваги паралельного виконання.	2	2
6.2 Робота з горутинами Вміє створювати, запускати та завершувати горутини. Знає поширені шаблони використання, такі як worker pool, fan-out/fan-in, та підходи до управління життєвим циклом горутин.	2	2
6.3 Синхронізація у Go Володіє навичками використання інструментів синхронізації, таких як м'ютекси, семафори та WaitGroup, для управління доступом до спільних ресурсів у багатопотоковому середовищі.	2	2
6.4 Комунікація через канали Знає, як використовувати канали для передачі даних між горутинами. Розуміє концепції буферизованих та небуферизованих каналів, а також практичні аспекти їх використання.	2	2
6.5 Практичні аспекти конкурентності Вміє вирішувати поширені проблеми конкурентності, такі як race conditions та deadlocks. Розуміє принципи оптимізації паралельного виконання та найкращі практики побудови конкурентних додатків.	2	2
7. Робота з мережею та створення REST API		
7.1 Основи роботи з мережею у Go Знає протоколи TCP/IP, HTTP/HTTPS і розуміє їхню реалізацію в Go. Вміє працювати з мережевими підключеннями, передавати дані та налаштовувати з'єднання.	3	3
7.2 Створення TCP-серверів Володіє навичками розробки серверів для обробки TCP-запитів. Розуміє, як налаштовувати з'єднання, обробляти клієнтські запити та передавати дані між клієнтом і сервером.	3	3
7.3 HTTP-сервери у Go Вміє налаштовувати простий HTTP-сервер, обробляти HTTP-запити та формувати відповіді. Розуміє основи роботи з маршрутизацією запитів.	3	3

7.4 Знайомство з фреймворком Fiber Ознайомлений із можливостями фреймворку Fiber, знає, як його встановлювати, налаштовувати та використовувати для розробки веб-додатків.	1	1
7.5 Розробка REST API з Fiber Вміє створювати REST API з використанням Fiber, налаштовувати маршрути, обробляти запити й відповіді, а також додавати обробку JSON-даних.	2	2
7.6 Практичне використання REST API Знає, як додати аутентифікацію, валідацію вхідних даних та обробку помилок у REST API, створених на основі Fiber.	2	2
7.7 Оптимізація та тестування API Розуміє принципи оптимізації продуктивності REST API, зокрема кешування та оптимізацію маршрутизації. Володіє інструментами для тестування, такими як Postman, та знає методи автоматизації тестування API.	2	2
8. Docker та бази даних		
8.1 Вступ до Docker Розуміє основи Docker, його ключові концепції, такі як образи, контейнери, Docker Hub, і переваги використання для контейнеризації додатків. Вміє пояснити, як Docker допомагає в розробці та розгортанні програмного забезпечення.	2	2
8.2 Налаштування Dockerfile Знає, як створювати Dockerfile для Go-додатків, визначати базові образи, налаштовувати середовище виконання, додавати залежності та оптимізувати розмір контейнерів шляхом використання багатостадійних збірок.	2	2
8.3 Управління контейнерами Вміє запускати, зупиняти, видаляти контейнери та працювати з Docker-образами. Розуміє команди для управління контейнерами (docker run, stop, rm, build, pull, push) і може перевіряти стан контейнерів.	2	2
8.4 Інтеграція з базами даних Ознайомлений із налаштуванням PostgreSQL та MongoDB у контейнерах Docker. Знає, як налаштувати мережеве підключення для взаємодії Go-додатків із базами даних, забезпечуючи безперебійну інтеграцію.	2	2
9. Git та управління версіями		
9.1 Основи Git Знає основи Git, як налаштувати локальні репозиторії, підключати їх до віддалених (GitHub, GitLab тощо). Має базове розуміння принципів версійного контролю, таких як коміти, гілки та злиття.	3	3
9.2 Основні команди Git Володіє навичками використання базових команд Git: clone для копіювання репозиторію, add для додавання змін, commit для фіксації змін, push та pull для синхронізації з віддаленим репозиторієм, status для перевірки стану репозиторію та log для перегляду історії змін.	3	3
9.3 Робота з гілками (Branches) Знає, як створювати та перемикатися між гілками в Git для організації ізольованих змін. Розуміє важливість гілок для паралельної розробки і оптимізації робочих процесів у команді.	3	3
9.4 Злиття та розв'язання конфліктів Вміє об'єднувати зміни з різних гілок за допомогою команди merge. Знає, як вирішувати конфлікти при злитті та уникати типових помилок при об'єднанні змін.	3	3
9.5 Практичні кейси роботи з Git Має досвід створення pull requests для внесення змін у основну гілку проекту, проведення рев'ю коду, а також співпраці у команді для забезпечення якісної та ефективної розробки.	3	3
10. Розмовна англійська		

10.1. Правильна побудова фрази з узгодженням часу Вміє використовувати відповідні часові форми дієслів і структури речень для точного вираження часу в комунікації. Здатний правильно використовувати часові форми, такі як Present Simple, Present Continuous, Past Simple, Past Continuous, Future Simple, для передачі правильного значення в повідомленні.	3	3
10.2 Професійна іт термінологія в розмові з замовником Знає і використовує професійні ІТ-терміни та скорочення, які використовуються в ІТ-індустрії, для чіткого та зрозумілого спілкування зі замовником. Вміє перекладати складні технічні концепції на доступну мову для замовника, не знайомого з ІТ-галуззю.	3	3
10.3 Професійна іт термінологія у діловому звіті Знає і використовує спеціалізовану ІТ-термінологію та фразеологію при написанні ділових звітів. Вміє передати інформацію про технічні питання та проєкти зрозуміло та конкретно, використовуючи адекватні ІТ-терміни та аббревіатури.	3	3
10.4 Професійна термінологія для Golang Dev Знає та розуміє специфічну термінологію, що використовується в контексті Golang Dev. Розуміє та використовує специфічні терміни, патерни, алгоритми та інші інструменти, які використовуються в Golang Dev	3	3
10.5 Неформальне спілкування (small talks) Вміє вести неформальну розмову, розпочинати діалоги та підтримувати невимушену атмосферу. Знає загальноживану термінологію та фрази, які використовуються у неформальних розмовах, таких як загальні питання про погоду, цікаві події, особисті захоплення, спорт, фільми тощо.	2	2

Коментарі до таблиці

Оцінки компетентностей вказані станом на 2024/2025 рр., по шкалі від 0 до 3.

«3» - компетентність вкрай важлива, якщо кандидат нею не володіє, його подальше просування на вакансію неможливе.

«2» - компетентність, знання якої обов'язково знадобиться кандидату при подальшому професійному зростанні.

«1» - некритична компетентність, якою можна знехтувати.

«0» - неактуальна компетентність.

Зауваження від компанії

SoftServe

1.5 - Важливо для Middle рівня

2.5 - Важливо для Middle рівня

2.6 - Важливо для Middle рівня

2.7 - Важливо для Middle рівня

3.5 - Важливо для Middle рівня

4.1 - Важливо для Middle рівня

4.2 - Перетинається з пунктом 3.3

6 - Важливо для Middle рівня

7.4 - Fiber буде корисним значно пізніше, для початку треба вивчити як все працює без нього

7.5 - Розробка REST API - 3

Fiber - 1

7.6 - Практичне використання REST API - 3

Fiber - 1

7.7 - Важливо для Middle рівня

8 - Важливо для Middle рівня