

ОПИС ПРОГРАМИ “Java Core”

Розділ 1. Java Core

- Модуль 1. Огляд мови Java, знайомство з Git.
 - Встановлення Java (JDK)
 - Встановлення IntelliJ IDEA
 - Створення проекту – hello world
 - Знайомство з git
 - Створення першого репозиторія GitHub
 - Налаштування git ignore
- Модуль 2. Змінні Java, введення/виведення через консоль.
 - Коментарі
 - Пристрій пам'яті в Java
 - Типи змінних (примітиви):
 - byte
 - char
 - short
 - int
 - float
 - double
 - long
 - Посилальні типи (String)
 - Оператори:
 - +, -, *, /
 - ++, --,
 - =, +=, -=
 - <,>, <=,>=, ==, !=
 - &&, ||, !,
 - %
 - Виведення інформації на консоль
 - Зчитування інформації з консолі
- Модуль 3. Масиви, цикли, умови та сортування.
 - Масив
 - Умовні конструкції: if, else, тернарний оператор, switch-case
 - Цикли: for, while, do-while, foreach
 - Ключове слово break
 - Ключове слово continue
 - Сортування у Java. Алгоритм "bubble sort"
- Модуль 4. ООП, класи, методи.
 - Принципи ООП
 - Метод та сигнатура методу
 - Модифікатори доступу
 - Getter/Setter
 - Class Object
 - Overriding
 - Overloading
 - Тип значення, що повертається (return, void)
 - Ключові слова abstract, final, static

- Модуль 5. Створення екземплярів класу, конструктор.
 - Конструктор класу
 - Ключові слова `this`, `super`
 - Блоки ініціалізації
 - Внутрішні класи
 - Винятки та їх ієрархія
 - `Checked/unchecked exception`
 - Обробка винятків (`try`, `catch`, `finally`, `throw`, `throws`)
 - Паттерни проектування
 - Принципи S.O.L.I.D.
 - GoF patterns: `Factory`, `Abstract Factory`
- Модуль 6. Робота з рядками. `Immutable objects`.
 - `Immutable / mutable object`
 - Рядкові класи: `String`, `StringBuilder`, `StringBuffer`
 - Розбиття та конкатенація рядків
 - Форматування рядків
 - Порівняння рядків
 - Можливості класу `String`
 - GoF patterns: `Builder`, `Command`, `Observer`
- Модуль 7. Інтерфейси. Колекції (інтерфейс `List`).
 - Інтерфейс
 - Анотація класів `vs` інтерфейси
 - `Collections`, `Map`
 - `interface List`:
 - `-- ArrayList`
 - `-- LinkedList`
 - `-- Stack`
- Модуль 8. Колекції (інтерфейси `Queue`, `Set`). `Map`.
 - Контракт `equals()`, `hashCode()`
 - `HashMap`
 - `interface Set`:
 - `-- HashSet`
 - `-- TreeSet`
 - `interface Comparable`
 - `Queue`
 - GoF patterns: `Iterator`
- Модуль 9. Робота з файлами. `Enum`. `Generic`.
 - `Enum`
 - Запис та зчитування з файлу:
 - `-- FileReader / FileWriter`
 - `-- FileInputStream / FileOutputStream`
 - `JSON`
 - `Serializable`
 - Ключове слово `transient`
 - `Generic` (узагальнений клас, метод)
- Модуль 10. Функціональне програмування `Java`. `Optional`.
 - Функціональні інтерфейси:
 - `-- Predicate`

- Consumer
 - Supplier
 - Function
- o Лямбда-вирази
- o Stream API
- o Optional
- Модуль 11. Багатопотоковість
 - o Процес vs потік
 - o Java Memory Model: принцип "happens-before"
 - o Створення нового потоку:
 - class Thread
 - interface Runnable
 - o Стан потоку
 - o Синхронізація потоків (volatile, synchronized)
 - o Монітори (wait, notify, notifyAll)
 - o Потоки-демони
 - o Deadlock
- Модуль 12. Concurrency.
 - o ExecutorService
 - o Класи-бар'єри:
 - Semaphore
 - CountdownLatch
 - CyclicBarrier
 - Phaser
 - Exchanger
 - o Atomic types (принцип CAS)
 - o Interface Lock
 - o Callable, FutureTask
 - o Fork/Join
- Модуль 13. Web, протоколи. Підсумки курсу.
 - o Інтернет-протоколи:
 - IP (маска, домен, DNS)
 - TCP / UDP
 - HTTPS / Web-socket
 - o Методи HTTP
 - o HTTP статуси
 - o REST vs SOAP
 - o Огляд додаткових можливостей мови:
 - JavaFX
 - Логування
 - Reflection API
 - ClassLoader**
 - Типи посилань (Reference)**

Розділ 2. Soft Skills

- Блок 1: Особиста ефективність
- Блок 2: Тайм-менеджмент
- Блок 3: Ефективна комунікація

	Блок 2. Java Developer	
	Модуль	Зміст
	Модуль 1. Системи збирання та управління залежностями	Що таке системи збирання та управління залежностями. Їх історія. Найпоширеніша система maven. Публічний каталог Maven Central Repository. Новітня система gradle. Її переваги. Обираємо між Maven і Gradle
	Модуль 2. Принципи тестування	Види та засоби тестування. TDD - задачі та переваги. Юніт-тестування з JUnit. Використання бібліотеки Mockito.
	Модуль 3. Якість коду	Що таке якість коду. Принципи чистого коду. Рефакторинг коду. Рев'ю коду. Стандарти кодування. Статичні аналізатори коду - SonarLint, Veracode.
	Модуль 4. Модель пам'яті Java	Модель пам'яті Java. Дослідження heapdump - мета та інструментарій. Поняття та огляд роботи різних типів GC. Анотації Java, приклад AOP. Використання JMX для спостереження за програмою.
	Модуль 5. Складність алгоритмів	Big(O) Динамічне програмування. Рекурсія. Алгоритми пошуку та сортування. Основи UML. Три-рівнева архітектура.

	Модуль 6. Огляд СУБД.	Реляційні та NoSQL БД, приклади застосування. Огляд H2, MySQL, PostgreSQL. Перша БД на прикладі MySQL. Принципи проектування БД. Тип даних, індекси та відношення між таблицями. Аналіз ефективності запитів на прикладі MySQL. Транзакції. Огляд збережених процедур та тригерів на прикладі MySQL.
	Модуль 7. Мова запитів SQL.	Діалект MySQL. Використання DDL, DML, DQL. Базові команди SQL. Групування та сортування. Агрегація, інші обчислення. Команди JOIN. Управління транзакціями. Віртуальні таблиці та вкладені запити.
	Модуль 8. Робота з БД через JDBC.	Огляд JDBC, підключення до БД. Концепція CRUD. Використання класів Entity. Виконання запитів за допомогою JDBC. Робота з транзакціями через JDBC. Використання бібліотеки Flyway.
	Модуль 9. Представлення даних з БД в Java.	Концепція ORM. Що таке JPA. Використання бібліотеки MyBatis.
	Модуль 10. Бібліотека Hibernate.	Огляд Hibernate. Принципова перевага Hibernate. Підключення до БД. Використання класів Entity в Hibernate, їхня ініціалізація та зв'язки між сутностями. Конфігурування моделі сутностей. Мова запитів HQL/HCQL, іменовані запити. Кешування в Hibernate. Робота з транзакціями через Hibernate.

	Модуль 11. Загальна архітектура мереж.	Історичний огляд. Основні компоненти мережи, доменні імена та IP-адреси. Архітектура клієнт-сервер та протокол HTTP(s). Загальні принципи побудови веб-додатків. Сервер веб-додатків Tomcat. Огляд технології сервлетів. Перший веб-додаток.
	Модуль 12. Огляд бібліотеки Spring.	Що таке Spring, його принципові переваги. Огляд складових частин. Переваги DI-контейнеру Spring. Відмінність Spring від Spring Boot. Що входить до Spring Core. Spring-компонент та його життєвий цикл. Настроювання конфігурації DI-контейнера. Перша програма на Spring Boot.
	Модуль 13. Spring MVC.	Що таке MVC. Реалізація Spring Web MVC. Структура веб-додатку за Spring MVC. JSP та Spring MVC Form Tags. Використання Spring MVC Validation. Приклад CRUD-сервісу згідно до Spring MVC. Використання бібліотеки Thymeleaf.
	Модуль 14. Spring Data.	Огляд Spring Data. Застарілий інтерфейс JdbcTemplate. Базовий інтерфейс Spring Data - Repository. Приклад реалізації власного Repository. Настроювання конфігурації Spring Data. Робота з транзакціями в Spring Data. Використання Hibernate разом із Spring. Порівняння JDBC, JPA, Hibernate та Spring Data.

	Модуль 15. Spring Security.	<p>Огляд функцій та переваг Spring Security.</p> <p>Побудова і робота компонентів Spring Security.</p> <p>Розгляд роботи компонентів Spring Security на прикладі функції "Remember me".</p> <p>Обмеження доступу до функцій за допомогою Spring Security.</p> <p>Безпека веб-додатків - веб-атаки, інекції, логування та зберігання паролів.</p> <p>Безпечне конфігурування java-додатків.</p>
	Модуль 16. Моніторинг веб-додатків.	<p>Цілі та методи моніторингу веб-додатків.</p> <p>Інтеграція з системою моніторингу Kibana.</p> <p>Особливості роботи з логами у веб-додатках.</p> <p>Перформанс-аналіз, використання JVisual VM</p> <p>Огляд принципів оптимізації роботи веб-додатків.</p> <p>Використання кешування в пам'яті для прискорення роботи веб-додатків.</p> <p>Використання CDN.</p>
	Модуль 17. Навантажувальне тестування веб-додатків.	<p>Цілі та види навантажувального тестування.</p> <p>Огляд популярних інструментів навантажувального тестування - JMeter, Curl, Postman, SoapUi.</p> <p>Використання Postman</p> <p>Використання JMeter.</p> <p>Використання інструменту Java Faker.</p>
	Модуль 18. Огляд REST API.	<p>Що таке RESTful-сервіси.</p> <p>Загальні принципи проектування API.</p> <p>Використання та переваги Swagger.</p> <p>Використання Spring для створення RESTful-сервісів.</p> <p>Особливості створення RESTful-сервісів та шляхи їх вирішення.</p>
	Модуль 19. Принципи AAA.	<p>Концепція AAA.</p> <p>Огляд OpenID та OAuth.</p> <p>Використання JWT, розробка token-based API.</p>

	Модуль 20. Інструменти розгортання.	Історія контейнеризації, різниця між Docker та Kubernetes. Використання Docker для локального розгортання веб-додатків. Огляд платформи Jenkins. Використання автоматизації разом з Git Actions. Огляд концепції CI/CD, стратегії розгортання в робочому середовищі.
--	-------------------------------------	--