

Mündəricat

Proqram dizaynı və C#.....	2
Seniorlar Üçün SQL (T-SQL)	5
Proqram dizaynı, Dizayn patternləri və dizayn prinsipləri.....	9
Domain-driven Design.....	16
TDD və Proqram təminatının test olunması (Unit və İntegration test).....	20
Distributiv sistemlər üçün Apache Kafka.....	21
Docker və Kubernetes	23
Mikroservis arxitekturası	25

Təlim: “Seniorluğa gedən yol”

Təlimçi : Tural Süleymani

Proqram dizaynı və C#

Məqsəd: Həlləri C# dilində verməklə, yekunda dildən asılı olmadan düşünmə vərdişləri qazanmaq. İstənilən çətinlikdə proqramları mümkün ola biləcək ən sadə yollarla, daha düzgün , genişlənə bilən, müşayiət olunan formada dizayn etmək vərdişləri qazanmaq.

Modul 1. C#/.NET-in daxili strukturu

- CLR və onun məsuliyyətləri, önəmi
- CIL, CLS, CTS, FCL,BCL və bunları bilməyin önəmi
- JIT Compiler və optimizasiya xüsusiyyətləri
- MSIL/IL dili nədir?
- MSIL/IL dilində praktiki proqramların yazılması
- Daxili infrastrukturun sonrakı dövrlərdə kod yazmaq prosesinə təsiri
- Assembly və dissassembly prosesləri, maşın dili səviyyəsində əməliyyatların yerinə yetirilməsi

Modul 2. Deleqatlar and Eventlar

- Deleqatlar reallıqda nədir, nə səbəbə mövcuddur?
- Yaxşı proqram dizaynında deleqatların yeri
- Real praktikada deleqatları bir Senior kimi istifadə edə bilmək
- Dizayn prinsipləri və deleqatlar
- Eventlar və proqram dizaynında onların yeri
- Dizayn prinsipləri və Eventlar
- Event vs Deleqat. Kimi harda istifadə edək?
- İnterfeyslər yoxsa deleqatlar?

Modul 3. C# və Funksional proqramlaşdırma

- Funksional proqramlaşdırma nədir? Funksional proqramlaşdırma və riyaziyyat
- Funksional proqramlaşdırmanın yaxşı proqram dizaynında rolu
- Primitive Obsession və proqram dizaynında rolu
- Railway Oriented Programming
- Real praktikada funksional proqramlaşdırma istifadə etməklə refactoring

Modul 4. Ümumiləşmələr, Funksional genişlənmələr və interfeyslər

- Ümumiləşmələr(Generics) və proqram dizaynında rolu
- Kovariantlıq, Kontravariantlıq, İnvariantlıq
- Funksional genişlənmələr(Method extensions) və Senior gözü ilə istifadəsi
- İnterfeyslər yoxsa abstrakt klaslar? Seniorlar interfeysləri necə müəyyən edir?
- Global proyektlər üzərindən adları sadalanan elementlərin real praktikada tətbiqi

Modul 5. C# və Concurrent Proqramlaşdırma

- Concurrent proqramlaşdırma və elementləri
- Concurrent proqramlaşdırmada thread, Thread Pool, Task və Paralellik
- Concurrent proqramlaşdırma problemləri və həll yolları
- Thread sinxronizasiya vahidləri
- Async/await və arxa planda necə işləmə incəlikləri
- Professional disassembly və State Machine
- Performans optimizasiyası və Asinxron proqramlaşdırma
- Real proyektlər üzərindən Concurrent proqramlaşdırmanın izahı
- Asinxron UI

Modul 6. C#-da Expression Tree-lər

- Expression Tree nədir və istifadə yerləri
- Real praktikada Expression Tree-lər
- Seniorlar üçün Expression Tree-lər, modulyar və dinamik sistemlərin yazılması

Modul 7. Reaktiv genişlənmələr

- Reaktiv proqramlaşdırma və məqsədi
- Reaktiv proqramlaşdırmanın real praktikada tətbiqi
- Reaktiv proqramlaşdırma ilə adaptiv kod dizaynı

Modul 9. Asp.net Core Web API ən yaxşı praktikaları

- Minimal API və kod dizaynı
- Asp.net Core Web API-da düzgün proqram strukturu

Modul 8. Bir Seniorun gözü ilə C#

- Real tapşırıq üzərindən 0-dan proqram yazılması
- Bir Seniorun gözü ilə proqramın review və refactor olunması
- Proqram təminatı və biznes
- C# nə qədər OOP dilidir?

Modul 9. Seniorların qəliz tapşırıqlara yanaşma qaydaları

- Daha yaxşı kod və yetəri qədər kod dizaynı
- Nəyi harada, necə, nəyə görə yazmalıyıq?
- Hara qədər genişlənən kod yazmalıyam?
- Abstrakt kod yoxsa birbaşa relizasiya. Dizaynda qızıl orta

Modul 10. Öyrənilənlərimizin praktiki təkrarı

Bu bölmədə indiyə kimi öyrəndiklərimizi real praktika tətbiq edəcəyik. Tələbələrin həllərini github özərindən yoxlayaraq onlara doğru və yanlışları haqqında ətraflı məlumat və mövbəti inkişaf mərhələləri haqda məsləhətlər veriləcək.

Seniorlar Üçün SQL (T-SQL)

Məqsəd: Həlləri SQL dilində verməklə, yekunda istənilən ,çətinlikdə biznes problemləri SQL dilində realizə edə bilmək verdişləri qazanmaq.

Modul 1. SQL arxa planda əslində necə qurulub?

- Sorğuların Konseptual İnterpretasiya Mərhələsi
- Konseptual interpretasiyanın gələcək SQL obyektlər dizaynında rolu
- Real Relyasiyalı model anlayışı
- Çoxluqlar nəzəriyyəsi və SQL
- Predikatlar məntiqi və SQL

Modul 2. SQL-in əsasları

- SELECT, aliaslar, identifikatorlar və obyektlərin düzgün adlanma formaları
- Where və kompleks filterlər
- Requlyar ifadələr
- SARG-a görə sorğuların yazılması
- FROM, GROUP BY, HAVING, ORDER BY , TOP , OFFSET..FETCH
- SQL və sadə alqoritmlərin SQL-də yazılması
- Kompleks qruplaşma sorğularının yazılması

Modul 3. Cədvəllərin düzgün dizayn olunması

- Cədvəl identifikator alqoritmləri və istifadə yerləri
- Foreign və primary key anlayışları
- Cədvəl Normalizasiya formaları
- Dəyişən tipləri, onların prioriteti və düzgün tiplərin təyin olunması
- Vaxt/ tarix və mətn tipli funksiyalar
- Cədvəllərarası əlaqələrin tələblər əsasında formalaşması
- Surraat açarlar
- Deterministik və deterministik olmayan funksiyaların yazılması

Modul 4. Join formaları , Apply və Pivot

- Karteziyan join və praktikada istifadəsi
- Inner join, Left join, right join, full join formaları
- Cross və Outer apply
- Join vs apply və praktikada istifadə yerləri
- Pivot cədvəllər və biznes keyslər

Modul 5. Subquery-lər və Cədvəl ifadələri(Table expressions)

- Subquery-lər , formaları və istifadə yerləri

- Dereived Table-lar əc CTE-lər, onları fərqi, istifadə yerləri
- Relyasiyalı model və Table expression-lar
- Rekursiyalar və yazılma formaları

Modul 6. Pəncərə funksiyaları(Windowed functions)

- Pəncərə funksiyaları və formaları
- Pəncərə funksiyaları və qruplaşdırma
- Pəncərə funksiyalarını növləri
- Kompleks pəncərə funksiyalarının yazılması

Modul 7. SQL və XML-lə iş

- Xml sənədlərin SQL-də saxlanması
- XML informasiya üzərində iş
- XML shredding
- XML manipulyasiya formaları və rela praktikada əhəmiyyəti
- XML tipi, Xquery, FLWOR və XPath

Modul 8. SQL və JSON-la iş

- JSON sənədlərin SQL-də saxlanması
- JSON informasiya üzərində iş
- JSON manipulyasiya formaları və real praktikada əhəmiyyəti

Modul 9. View-lar və Funksiyalar

- View və yaradılma formaları

- View və istifadə yerləri
- Funksiyalar, onların formaları (inline, Table-valued, multi-statement)
- Funksiyalar və proqram təminatında istifadə yerləri

Modul 9. Stored Procedure-lar

- Stored prosedurlar və proqram təminatında istifadə yerləri
- Profesional stored prosedurların yazılması
- Biznes tələblərin prosedurlarda əksi

Modul 10. SQL və Transaksiyalar

- Tranzaksiyaların formaları və tətbiq yerləri
- Transaksiya səviyyələri və istifadə yerləri
- ACID prinsiplər və transaksiyalar
- Transaksiya əsaslı stored prosedurların yazılması
- Biznes tələblərin transaktiv mühitdə təsviri

Modul 11. Öyrənilənlərin praktiki təkrarı

Bu bölmədə indiyə kimi öyrəndiklərimizi real praktika tətbiq edəcəyik. Bütöv bir biznes tələblər toplusu SQL-də həll edilməklə, çətinlik dərəcəsi ort-yuxarı olan tapşırıq verilir. Tələbələrin həllərini github özərindən yoxlayaraq onlara doğru və yanlışları haqqında ətraflı məlumat və mövbəti inkişaf mərhələləri haqda məsləhətlər veriləcək.

Proqram dizaynı, Dizayn patternləri və dizayn prinsipləri

Məqsəd: Bir Seniorun, gündəlik həyatda düzgün, genişlənən, rahat müşayiət olunan kod yazma prosesində istifadə etdiyi dizayn qaydaları, prinsipləri və patternlərini mənimsəmək və REAL PRAKTİKAdə onların tətbiq yerləri bilmək, bir-birindən fərqləndirmək. Hansı patterni harada və nəyə görə istifadə edim kimi suallara cavab tapmaq.

Modul 1. İrəli səviyyə Obyektyönümlü proqramlaşdırma(OYP)

- İnkapsulyasiya və informasiyanın gizlədilməsi
- Törəmə : pattern yoxsa antipattern
- Polimorfizm və onun formaları
- Dispatching və onun formaları, polimorfizmlə əlaqəsi
- Realda abstraksiya nədir?
- İndiki proqramçılar nəyə görə bunları səhv bilir?
- Əslində bir dilin OYP olması üçün şərtlər

Modul 2. İrəli səviyyə SOLİD prinsiplər

- SRP və hərəkəsin bildiyi yalnışlar
- OCP və bilinməyən tərəfləri
- Əslində LSP nəyə görə onların bildiyi LSP deyil
- Interface Segregation və proqram dizaynı
- Dependency Inversion və onun Dİ və konteynerlərlə əlaqəsi

Modul 3. Proqram arxitekturasında high cohesion və loosely coupling

- Proqram dizaynı əslində haradan başlayır?
- High cohesion nədir və proqram dizaynında rolu
- Loosely coupling nədir və proqram dizaynında rolu
- High cohesion formaları və istifadə yerləri
- Loosely coupling formaları və istifadə yerləri

Modul 4. Dizayn patternlərinə giriş

- Dizayn patternləri əslində nədir?
- Proqramlaşdırmada gümüş güllə (Silver bullet)
- İstifadə etmək ya eyməmək? bunu necə müəyyən edək?
- Patternlərin tarixi və müasir proqramlaşdırma

Modul 5. Strategiya patterni

- Strategiya patterni nədir və nəyə xidmət edir?
- Strategiya bir fəlsəfə kimi
- Bilərək və bilmədən indiyə kimi haralarda istifadə etmişik?
- Strategiya və abstraksiya
- İnterfeys yoxsa deleqat
- Bir Senior bu patternin istifadə olunub olunmayacağını necə müəyyən edir?
- .NET-də istifadə yerləri
- Harada və nə zaman istifadə etməliyik?

Modul 6. Mediator patterni

- Mediator patterni nədir və nəyə xidmət edir?
- Bilərək və bilmədən indiyə kimi haralarda istifadə etmişik?
- Mediator və real həyat fəlsəfəsi
- Arxitektut mediatorlar
- .NET-də istifadə yerləri
- Harada və nə zaman istifadə etməliyik?

Modul 7. Template method patterni

- Template method patterni nədir və nəyə xidmət edir?
- Bilərək və bilmədən indiyə kimi haralarda istifadə etmişik?
- Template method və törəmə probleminin həlli
- Template method hansı 3 formada realizə olunur?
- Template method yoxsa strategiya patterni?
- Harada və nə zaman istifadə etməliyik?

Modul 8. Iterator patterni

- İterator patterni nədir və nəyə xidmət edir?
- Bilərək və bilmədən indiyə kimi haralarda istifadə etmişik?
- İterator, real GOF təsviri və C#-a adaptasiyası
- .NET-də iterator interfeyslər
- Internal və external iteratorlar
- Harada və nə zaman istifadə etməliyik?

Modul 9. Command patterni

- Komanda patterni nədir və nəyə xidmət edir?
- Bilərək və bilmədən indiyə kimi haralarda istifadə etmişik?
- Real həyatda komanda patterni

Modul 10. Observer patterni

- Observer patterni nədir və nəyə xidmət edir?
- Bilərək və bilmədən indiyə kimi haralarda istifadə etmişik?
- Pull və Push əsaslı Observerlər
- .NET-də Observerlərin 4 realizasiya formaları
- Reaktiv programlaşdırma və Observer patterni
- Harada və nə zaman istifadə etməliyik?

Modul 11. Visitor patterni

- Visitor patterni nədir və nəyə xidmət edir?
- Bilərək və bilmədən indiyə kimi haralarda istifadə etmişik?
- Visitor və dispatching
- Harada və nə zaman istifadə etməliyik?

Modul 12. State patterni

- State patterni nədir və nəyə xidmət edir?
- State machine nədir?
- .NET-də realizasiya forması

Modul 13. Factory method patterni

- Factory method patterni nədir və nəyə xidmət edir?
- Bilərək və bilmədən indiyə kimi haralarda istifadə etmişik?
- 3 realizasiya forması
- Nə zaman, hansı realizasiya formasını istifadə etməliyəm?

Modul 14. Chain of Responsibility patterni

- CoR patterni nədir və nəyə xidmət edir?
- Bilərək və bilmədən indiyə kimi haralarda istifadə etmişik?
- .NET-də realizasiya formaları
- Harada və nə zaman istifadə etməliyik?

Modul 15. Singleton patterni

- Singleton patterni nədir və nəyə xidmət edir?
- Bilərək və bilmədən indiyə kimi haralarda istifadə etmişik?
- .NET-də realizasiya formaları
- Singleton vs static
- Harada və nə zaman istifadə etməliyik?

Modul 16. Abstract Factory patterni

- Abstract Factory patterni nədir və nəyə xidmət edir?
- İndiyə kimi haralarda istifadə etmişik?
- Realizasiya forması
- Real paraktikada harada istifadə edək?

Modul 17. Builder patterni

- Builder patterni nədir və nəyə xidmət edir?
- İndiyə kimi haralarda istifadə etmişik?
- Əsas 2 realizasiya forması
- Builder vs immutability
- .NET-də builderlər
- Real paraktikada harada istifadə edək?

Modul 18. Adapter patterni

- Adapter patterni nədir və nəyə xidmət edir?
- İndiyə kimi haralarda istifadə etmişik?
- Class vs object adapter
- Əsas 2 realizasiya forması
- Adaptiv refactoring
- Real paraktikada harada istifadə edək?

Modul 19. Facade patterni

- Facade patterni nədir və nəyə xidmət edir?
- İndiyə kimi haralarda istifadə etmişik?
- .NETdə facadelər
- Real paraktikada harada istifadə edək?

Modul 20. Decorator patterni

- Decorator patterni nədir və nəyə xidmət edir?
- İndiyə kimi haralarda istifadə etmişik?
- Composition vs Inheritance
- Real praktikada harada istifadə edək?

Modul 21. Composite patterni

- Composite patterni nədir və nəyə xidmət edir?
- İndiyə kimi haralarda istifadə etmişik?
- Composition vs Inheritance
- Praktikada harada istifadə edə bilərik?

Modul 22. Flyweight patterni

- Flyweight patterni nədir və nəyə xidmət edir?
- İndiyə kimi haralarda istifadə etmişik?
- Praktikada harada istifadə edə bilərik?

Modul 23. Bridge patterni

- Bridge patterni nədir və nəyə xidmət edir?
- İndiyə kimi haralarda istifadə etmişik?
- Praktikada harada istifadə edə bilərik?

Modul 24. Prototype patterni

- Prototype patterni nədir və nəyə xidmət edir?
- İndiyə kimi haralarda istifadə etmişik?

Modul 25. Dizayn patternlərinin yekunu

- Digər dizayn patternləri hansıdır?
- Dizayn patternlərinə necə yanaşmaq lazımdır?
- Hansı hallarda pattern tətbiq etmək haqda düşünmək lazımdır?
- Patternlər və anti-patternlər. Patternlər necə anti-patternə çevrilir?

Modul 26. Öyrənilənlərimizin praktiki təkrarı

Bu bölmədə indiyə kimi öyrəndiklərimizi real praktika tətbiq edəcəyik. Bütöv bir proqramın müxtəlif patternlər və prinsiplər tətbiq etməklə, çətinlik dərəcəsi orta-yuxarı olan tapşırıq verilir. Tələbələrin həllərini github özərindən yoxlayaraq onlara doğru və yanlışları haqqında ətraflı məlumat və mövbəti inkişaf mərhələləri haqda məsləhətlər veriləcək.

Domain-driven Design

Məqsəd: DDD-ni real praktikada necə tətbiq edə bilərik? DDD-nin stratejik və taktikal patternləri hansılardır? Xarici şirkət DDD-ni real praktikada necə tətbiq edir? Bu sualları cavablandırmağın yanı sıra, bu modullar prosesində tam real praktikanı əks etdirən DDD proyektini sıfırdan hazırlayacağıq və hansı patternləri və prinsipləri nə zaman istifadə edəcəyimizi birlikdə öyrənəcəyik.

Modul 1. Giriş

- DDD nədir?
- DDD hansı patternlər təqdim edir?
- DDD və biznesin əlaqəsi nədir? DDD biznesə necə xidmət edir?
- Hansı proyektlərə DDD tətbiq edilməlidir?
- Anemic model problemi və Rich domain model

Modul 2. Ubiquitous language (UL)

- UL nədir və DDD-də əhəmiyyəti
- UL necə müəyyən olunur?
- Komandanın UL müəyyənləşdirilməsində rolu

Modul 3. Domen, subdomen və Bounded Context

- Domen, subdomen və Bounded context nədir?
- Problem space və solution psace
- Kenevin framework və tətbiq forması
- Bounded Context necə müəyyənləşdirilir?
- Subdomen və Bounded Context. Hansına nə zmaan fokuslanaq?

Modul 4. Context Mapping (CM)

- CM nədir?
- CM-in hansı formaları var?
- DDD-də organizational və integration patternləri
- Shared Kernel nədir? Hansı hallarda istifadə olunmalıdır?
- CM-in əməliyyəti və DDD-ə yeri

Modul 5. DDD və arxitektura

- Layered arxitektura və DDD-ə yeri
- Clean arxitektura və DDD-ə yeri
- Heksaqonal arxitektura və DDD-ə yeri
- Event-driven arxitektura və DDD-ə yeri

Modul 6. Entity-lər, Value Object-lər

- DDD-də Entity-lərin əhəmiyyəti
- Entity-lər necə müəyyən olunur?
- Entity-lər necə dizayn olunur?
- DDD-də Value Object-lərin əhəmiyyəti
- Value Object-lər necə müəyyən olunur?
- Value Object-lər necə dizayn olunur?

Modul 7. Factories(Fabrikalar) və Repository-lər

- DDD-də Fabrikaların əhəmiyyəti
- Nə zaman fabrika tətbiq etməliyik nə zaman yox?
- Real projektlərdə repositorylər harada tətbiq etmişəm?
- DDD-də Repositorylərin əhəmiyyəti
- Nə zaman repository tətbiq etməliyik nə zaman yox?
- Real projektlərdə repositorylər harada tətbiq etmişəm?

Modul 8. Aqreqatlar və Domain Event-lar

- DDD-də Aqreqat-lərin əhəmiyyəti
- Aqreqat-lar necə müəyyən olunur?
- Aqreqat-lar necə dizayn olunur?
- DDD-də Domain Event-lərin əhəmiyyəti
- Domain Event-lər necə müəyyən olunur?
- Domain Event-lər necə dizayn olunur?

Modul 9. Event Sourcing və Event-driven Arxitektura

- Event Sourcing nədir və DDD-də əhəmiyyəti
- Event sourcing dəstəkləyən arxitekturanın təmin olunması
- Event sourcing və klassik tranzaktiv yanaşma

Modul 10. DDD və real praktika

- DDD patternləri real praktikada necə tətbiq edilməlidir?
- Mən öz praktikamda hansı şirkətlərdə, necə tətbiq etmişəm?
- DDD tətbiqi zamanı diqqət etməli olduğumuz məqamlar
- DDD və mikroservis arxitekturası

TDD və Proqram təminatının test olunması (Unit və Integration test)

Məqsəd: TDD nədir və TDD əsaslı proqram ətminatı necə yazılır? Hansı hissələrə, necə profesional testlər yazı bilərik? İnteqrasiya testləri və unit testlər necə yazılmalıdır? Bu sualların cavabını bu bölmədə tapacağıq.

Modul 1. Unit test və texnikaları

- Unit test nədir? nə zaman yazılmalıdır?
- Sadə unit testlər necə yazılır?
- Proqram arxitekturası testə icazə vermədikdə

Modul 2. Unit test konseptləri

- Stub nədir və nə zman işlədək?
- Fake nədir və əhəmiyyəti
- Moq nədir və praktiki istifadəsi

Modul3. İrəli səviyyə Unit testing

- Fixture nədir və əhəmiyyəti
- The Collection Fixture
- Testlərin paralel işə salınması

Modul 4. Test-driven Development (TDD)

- TDD nədir?
- Real praktikada TDD necə tətbiq olunur?
- TDD tətbiq edərkən nəzərə almalı olduğumuz məqamlar
- TDD və digər proqram metodikalarının müqayisəsi

Modul 5. İnteqrasiya testləri və texnikaları

- İntegration test nədir? nə zaman yazılmalıdır?
- Sadə integration testlər necə yazılır?
- Proqram arxitekturası testə icazə vermədikdə

Modul 6. İnteqrasiya testləri və praktika

- Status kodların test olunması
- JSON cavabların testi
- Test datalar və onlarla iş
- Asp.NET Core Web API üçün integration testlər

Distributiv sistemlər üçün Apache Kafka

Məqsəd: A-dan Z-yə bütün Apache Kafka konseptlərini profesional səviyyədə nəzərdən keçirəcəyimiz bu bölmədə , Apache Kafkanın distributiv sistemlərdə əhəmiyyəti və praktikada tətbiq olunmasından danışacağıq.

Modul 1. Apache Kafkaya giriş

- Apache Kafka nədir?
- Distributiv sistemlərdə axtardığımız atributlar
- Kafka atributlara malikdir?
- Kafka-nın müxtəlif yüklənmə formaları

Modul 2. Apache Kafkanın əsasları

- Cluster və broker
- Topic və partition anlayışları
- Kafka Broker discovery
- Kafka CLI ilə tanışlıq
- Leader Partition
- Kafka consumers replicac fetching
- Topic replication factor

Modul 3. Kafka Producer

- Producer nədir?
- Producer message internals
- Kafka CLİ ilə Producer
- Producer Mesajlarının serializasiyası
- Partition strategiyaları
- Producer ack

Modul 4. Kafka Consumer və Consumer group

- Consumer nədir?
- Consumer mesajlarının deserializasiyası
- Kafka CLİ ilə Consumer
- Consumer Group və praktikada istifadəsi
- Offsetlərlə iş

Modul 5. İrəli səviyyə Kafka

- Partition distribution strategiyaları
- Distributiv konsensus protokolu
- Rebalans strategiyaları
- Statik qrup üzvlüyü
- Kompres olunma formaları

Modul 6. Real praktikada Kafka və .NET

- Producer və Consumer sistemini qurulması
- .NET və Kafka ilə müştərək iş

- Yüksək sürətli producerlərin yazılması
- Producerlərin idempotantlığı

Docker və Kubernetes

Məqsəd: A-dan Z-yə bütün Docker və Kubernetes konseptlərini profesional səviyyədə nəzərdən keçirəcəyimiz bu bölmədə , proqramların docker ilə konteynerizə olunması və kuberneterlə deploy olunmasına baxacağıq.

Modul 1. Docker-ə giriş

- Docker nədir?
- Dockerə niyə ehtiyac var?
- Image və konteyner anlayışları
- Linuxsun əsasları: namespace və control group anlayışları

Modul 2. Dockerlə iş

- Run, create, start, ps, stop, kill, logs, exec, shell access komandaları
- Dockerfile yaradılması və lifecycle
- Taglər və onlarla iş
- Manual image yaradılma forması

Modul 3. Real node.js proqramın dockerize olunması

- Real proqramları necə dockerizə edə bilərik?
- Dockerfile-in icəlikləri
- Bir neçə servisin docker-compose ilə strukturlaşdırılması
- Azure Cloud-a deploy olunma

Modul 4 . Kuberneteslə iş

- Kubernetes nədir?
- Kuberneteslə işləmə forması və komandalar
- Kubernetesdə multi konteynerlər
- Ingress kontroller və trafik
- Production üçün deployment
- Kubernetes və HTTPs setup
- Skaffold ilə lokal deployment

Mikroservis arxitekturası

Məqsəd: Mikroservislərin real praktikada necə tətbiq edə bilərik? Mikroservis patternləri hansılardır? Xarici şirkət Mikroservisləri real praktikada necə tətbiq edir? Bu sualları cavablandırmağın yanı sıra, bu modullar prosesində tam real praktikanı əks etdirən mikroservis proyektini sıfırdan hazırlayacağıq və hansı patternləri və prinsipləri nə zaman istifadə edəcəyimizi birlikdə öyrənəcəyik.

Modul 1. Mikroservislərə giriş

- Mikroservis arxitekturası nədir?
- Software Development Lifecycle prosesi
- Big ball of Mud və program arxitekturasının əhəmiyyəti
- Monolit arxitektura: üstün və mənfi tərəfləri
- Servis əsaslı arxitektura: üstün və mənfi tərəfləri
- Mikroservis arxitektura: sütün və mənfi tərəfləri

Modul 2. Mikroservis arxitekturasının müəyyən olunması

- Tələblərin analizi və arxitekturaya təsiri
- Hansı hallarda mikroservis arxitekturası seçilməlidir?
- Mikroservis arxitekturasının minimal tələbləri
- Bounded Context və mikroservislər

Modul 3. Vertical-slice arxitektura ilə mikroservislərin yazılması

- Vertical-slice arxitektura nədir? Hansı hallarda istifadə olunmalıdır?
- Vertical-slice arxitektura ilə real praktikada mikroservis hazırlanması

Modul 4. gRPC ilə mikroservislərin yazılması

- RPC arxitektura nədir? Hansı hallarda istifadə olunmalıdır?
- gRPC ilə real praktikada mikroservis hazırlanması

Modul 5. Clean arxitektura ilə mikroservislərin yazılması

- Clean arxitektura nədir? Hansı hallarda istifadə olunmalıdır?
- Vertical-slice arxitektura ilə real praktikada mikroservis hazırlanması

Modul 6. Mikroservislərarası sinxron əlaqənin təmin olunması

- Mikroservis kommunikasiyasının əsasları
- Mikroservislərarası sinxron kommunikasiyanın qurulması

Modul 7. Mikroservislərarası asinxron əlaqənin təmin olunması

- Asinxron kommunikasiyanın əsasları
- Mikroservislərarası asinxron kommunikasiyanın qurulması
- Apache Kafka ilə asinxronluq

Modul 8. Mikroservislərarası real-time streaming

- Real-time kommunikasiyanın əsasları
- Mikroservislərarası streaming qurulması

Modul 9. Mikroservislərin təhlükəsizliyi

- Mikroservislərdə avtorizasiya və autentifikasiya
- Təhlükəsizlik üçün ən yaxşı praktikalar
- Təhlükəsizliyin təmin olunması

Modul 10. Mikroservislərin monitoring olunması

- Observability və əhəmiyyəti
- Loqlama üçün ən yaxşı praktikaalr və struktur loqlama
- ELK stack üçün monitoringin təmin olunması

Modul 11. Mikroservislər arxitekturası

- API Gateway
- CQRS və Event sourcing
- SAGA pattern

Modul 12. Mikroservislərin test olunması

- Mikroservislər üçün unit testlər necə yazılmalıdır?
- Mikroservislər üçün integrasiya testlərinin yazılması

Modul 13. CI/CD pipeline və mikroservislərin Cloud-a deploy olunması

- CI və CD nədir?
- Github Actions üzərindən pipeline necə qurulur?
- Mikroservisləri Cloudə necə deploy etmək olur?