

Mündəricat

Proqram dizaynı, Dizayn patternləri və dizayn prinsipləri	2
Domain-driven Design	10
System Design	14
TDD və Proqram təminatının test olunması (Unit və İntegration test)	17
Mikroservis arxitekturası	19

Təlim: “Arxitektorun yolu”

Təlimçi : Tural Süleymani

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 PRAKTİKADA 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)

- Inkapsulyasiya 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 yalnızlar
- 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?
- .NET-də facadelar
- 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.

Sistem Dizaynı

Məqsəd: Amazon, Google, Microsoft kimi big-tech şirkətlərdə sistem dizaynı ilə bağlı veriləcək sualları tam şəkildə əhatə etmək, praktiki nümunələrlə sistem dizaynını öyrənmək.

Modul 1. Sistem Dizaynına Giriş

- Sistem dizaynı mü sahibələrinin məqsədi və əhəmiyyəti
 - High-level design (HLD) və Low-level design (LLD) fərqləri
 - Design prosesində tələblərin toplanması və sualların aydınlaşdırılması
 - Müsahibə zamanı cavab strukturlaşdırma texnikaları
-

Modul 2. Genişlənəbilərlik və Yük Paylanması

- Horizontal vs vertical scaling
 - Load balancer-lərin rolu (Round Robin, Least Connections, Consistent Hashing)
 - Stateless və stateful servislər
 - Auto-scaling strategiyaları
-

Modul 3. Şəbəkə Əsasları və Protokollar

- TCP, UDP və fərqləri

- HTTP/HTTPS, gRPC, WebSockets istifadəsi
 - CDN konsepti və istifadə halları
 - API Gateway dizaynı
-

Modul 4. Data Storage Sistemləri

- Relational DB vs NoSQL DB (SQL, MongoDB, Cassandra, DynamoDB)
 - CAP teoremi və tətbiqləri
 - Partitioning (sharding) və replication
 - Database indexing və query optimizasiyası
-

Modul 5. Cache və Performance Optimallaşdırma

- Cache nədir və niyə istifadə olunur?
 - Cache strategiyaları (write-through, write-back, write-around)
 - Cache eviction strategiyaları (LRU, LFU, FIFO)
 - CDN və edge caching
-

Modul 6. Mesajlaşma və Event-Driven Architecture

- Message Queue (Kafka, RabbitMQ, Azure Service Bus)
 - Pub/Sub pattern
 - Event sourcing və CQRS giriş
 - Idempotency və at-least-once vs exactly-once processing
-

Modul 7. Consistency, Availability və Partition Tolerance

- CAP teoremi detallı
 - Strong consistency vs eventual consistency
 - Quorum-based yazma/oxuma əməliyyatları
 - Real nümunələr (Amazon Dynamo, Google Spanner)
-

Modul 8. Fault Tolerance və Reliability

- Failover və replication strategiyaları
 - Heartbeat, leader election (Raft, Paxos)
 - Retry pattern, circuit breaker pattern
 - SLA, SLO, SLI anlayışları
-

Modul 9. Distributed Systems Konsepsiyaları

- Consensus problem (Raft, Paxos)
 - Gossip protokolu
 - Time synchronization (NTP, logical clocks, vector clocks)
 - Distributed transactions və 2-phase commit
-

Modul 10. Sistem Dizayn Nümunələri (I)

- URL shortener dizaynı
- Rate limiter dizaynı
- Notification service dizaynı
- Key-value store dizaynı

Modul 11. Sistem Dizayn Nümunələri (II)

- Instagram feed dizaynı
- WhatsApp/Telegram messaging sistemi
- YouTube video streaming sistemi
- Uber/Lyft ride-hailing sistemi

Modul 12. Sistem Dizayn Nümunələri (III)

- Amazon “Add to Cart” sistemi
- Google Docs collaborative editing
- Twitter timeline və fan-out dizaynı
- Netflix recommendation engine overview

Modul 13. Monitoring və Observability

- Logging, metrics, tracing (ELK, Prometheus, Grafana, OpenTelemetry)
- Alerting və anomaly detection
- Health checks və heartbeats
- Chaos engineering

Modul 14. Security və Authentication

- Authentication və authorization (OAuth2, JWT, SAML)
- Data encryption (at rest, in transit)

- DDOS protection və rate limiting
 - Zero trust architecture
-

Modul 15. Cloud və Deployment Strategiyaları

- Monolith vs microservices vs serverless
 - Containerization və orchestration (Docker, Kubernetes)
 - Multi-region və multi-cloud deployment
 - Blue-green və canary deployment strategiyaları
-

Modul 16. Müsahibə Hazırlığı və Təcrübə

- Sistem dizayn suallarını cavablandırma metodologiyası
- Whiteboard və on-site müsahibələrdə yanaşmalar
- Mock interview practice (Amazon, Google, Microsoft nümunələri)
- Resurslar: kitablar, bloglar, LeetCode/System Design Primer

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 fokuslanmaq?

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 İntegration test)

Məqsəd: TDD nədir və TDD əsaslı proqram təminatı necə yazılır? Hansı hissələrə, necə profesional testlər yazıla bilər? İntegrasiya 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 zaman 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

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 proqram arxitekturasının əhəmiyyəti
- Monolit arxitektura: üstün və mənfi tərəfləri
- Servis əsaslı arxitektura: üstün və mənfi tərəfləri
- Mikroservis arxitektura: üstün 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?