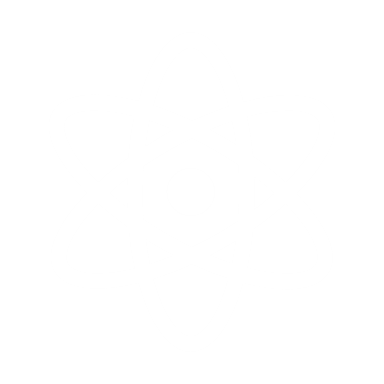
|  |  |
| --- | --- |
| **8-MAVZU** | **Domain-Driven Design (DDD) arxitektura asosidagi tizim yaratish.** |

****

**AMALIY  
KO‘NIKMA**

**HAR BIR TALABA O‘ZINI VARIANTIGA MOS MISOLNI ISHLAB VA OLINGAN NATIJALARNI BAHOLASH UCHUN PDF FORMATGA O‘TQAZIB <https://elms.tuit.uz> SAYTIGA YUKLASHI LOZIM**

**HAR BIR TALABA O’ZI TANLAGAN IXTIYORIY MAVZUDAGI LOYIHASINI USHBU TOPSHIRIQLAR ASOSIDA BAJARISHI TALAB QILINADI.**

**TANLANGAN LOYHA MAVZULARI BIR XIL BO’LMASLIGI KERAK AGAR BIR XIL LOYHADA BIR HIL ISH YUKLANSA BALL QO‘YILMAYDI HAMDA BAJARILGAN ISH DASTURDAN SKRINSHOT OLIB NATIJALARI JOYLANISHI TALAB QILINADI.**

**Maqsad**: Ushbu amaliy mashg‘ulotning maqsadi **Domain-Driven Design (DDD)** arxitekturasi asosida dasturiy tizim yaratish tamoyillarini tushunish va qo‘llashdan iborat. Talabalar DDD konsepsiyasining asosiy tushunchalarini (Domain, Entity, Value Object, Aggregate, Repository, Service) o‘rganadilar va ushbu yondashuv asosida tizimni loyihalash va dasturlash ko‘nikmalarini shakllantiradilar.

**Topshiriq variantlari**

1. **Domen modelini aniqlash**

* Belgilangan tizim uchun asosiy Entity, Value Object, va Aggregate larni aniqlang.
* Masalan, Customer, Product, va Order obyektlarini yarating.

1. **Value Object yaratish**

* Address, Price kabi o‘zgarmas (immutable) qiymat obyektlari ni @dataclass(frozen=True) yordamida yarating.

1. **Entity yaratish**

* Customer, Product, va Order sinflarini \_\_init\_\_ metodi orqali atributlari bilan birga yozing.
* Har bir entitet uchun UUID orqali noyob identifikator yarating.

1. **Agregatni yaratish va boshqarish**

* Order agregati ichida Customer va Product obyektlarini boshqarish imkoniyatini qo‘shing.
* Buyurtmaga mahsulot qo‘shish, mahsulot miqdorini tekshirish funksiyalarini yozing.

1. **Repository (Ma’lumotlar Ombori) yaratish**

* Buyurtmalarni saqlash uchun OrderRepository sinfini yarating.
* Buyurtmalarni saqlash, qidirish, va ro‘yxatni chiqarish metodlarini yozing.

1. **Xizmatlar (Services) qo‘shish**

* OrderService sinfini yarating.
* Buyurtma berish (place\_order), buyurtmani tasdiqlash (complete\_order) metodlarini yozing.

1. **Dasturga yangi xususiyat qo‘shish**

* Buyurtmalarga “Bekor qilish” (cancel\_order) funksiyasini qo‘shing.
* Buyurtma holatini Pending, Completed, Cancelled kabi holatlarga ajrating.

1. **Test ma’lumotlari bilan ishlash**

* Turli mahsulotlar, mijozlar va buyurtmalar yaratib, tizimni sinab ko‘ring.
* Masalan, order\_service.place\_order() orqali yangi buyurtma yaratib, uning natijasini chop eting.

1. **Ma’lumotlarni JSON formatida saqlash va yuklash**

* Buyurtmalarni JSON formatda faylga saqlash va yuklash funksiyalarini qo‘shing.
* json.dump() va json.load() funksiyalaridan foydalaning.

1. **Tizimni kengaytirish (Advanced)**

* Buyurtma holatini boshqa xizmatlar (notification, payment) bilan bog‘lash uchun Event-Driven Architecture dan foydalaning.
* Asinxron (async) ishlash uchun asyncio yoki threading kutubxonasidan foydalaning.

**Xulosa:**

Ushbu 10 ta amaliy topshiriq orqali DDD arxitekturasi asosida tizim yaratish va uni real biznes jarayonlari bilan bog‘lash ko‘nikmalariga ega bo‘lasiz.