КПО дз2
Балдина Анастасия, БПИ235

Реализованный функционал:

- **Добавить / удалить животное**
  1. Presentation
     1.1. Контроллер AnimalController
     1.2. Фасадный сервис AnimalFacadeServiceImpl
  2. Application
     2.1. Бизнес-сервис AnimalServiceImpl
  3. Domain
     3.1. Агрегат AnimalAggregate
  4. Infrastructure
     4.1. Репозиторий InMemoryAnimalRepository
- **Добавить / удалить вольер**
  1. Presentation
     1.1. Контроллер EnclosureController
     1.2. Фасадный сервис EnclosureFacadeServiceImpl
  2. Application
     2.1. Бизнес-сервис EnclosureServiceImpl
  3. Domain
     3.1. Агрегат EnclosureAggregate
  4. Infrastructure
     4.1. Репозиторий InMemoryEnclosureRepository
- **Переместить животное между вольерами**
  1. Presentation
     1.1. Контроллер EnclosureController
     1.2. Фасадный сервис EnclosureFacadeServiceImpl
  2. Application
     2.1. Бизнес-сервис AnimalTransferServiceImpl
  3. Domain
     3.1. Агрегат EnclosureAggregate
  4. Infrastructure
     4.1. Репозиторий InMemoryEnclosureRepository
     4.2. Репозиторий InMemoryAnimalRepository
- **Просмотреть расписание кормления**
  1. Presentation
     1.1. Контроллер FeedingFacadeController
     1.2. Фасадный сервис FeedingFacadeServiceImpl
  2. Application
     2.1. Бизнес-сервис FeedingOrganizationServiceImpl
  3. Domain
     3.1. Агрегат FeedingScheduleAggregate
  4. Infrastructure
     4.1. Репозиторий InMemoryFeedingRepository
- **Добавить новое кормление в расписание**
  1. Presentation
     1.1. Контроллер FeedingFacadeController

1.2. Фасадный сервис FeedingFacadeServiceImpl
2. Application
   2.1. Бизнес-сервис FeedingOrganizationServiceImpl
3. Domain
   3.1. Агрегат FeedingScheduleAggregate
4. Infrastructure
   4.1. Репозиторий InMemoryFeedingRepository

- **Просмотреть статистику зоопарка (кол-во животных, свободные вольеры и т.д.)**
  1. Presentation
     1.1. Контроллер StatisticsFacadeController
     1.2. Фасадный сервис StatisticsFacadeServiceImpl
  2. Application
     2.1. Бизнес-сервис ZooStatisticsServiceImpl

Концепции Domain-Driven Design

- **Ограниченный контекст**
  1. На каждом слое (Presentation,Application, Domain, Infrastructure) используется несколько моделей - animal, enclosure, feeding, statistics. В проекте это разбиение реализовано с помощью Java package.
  2. Доменные модели полностью изолированы друг от друга:
     2.1. Модель Animal: сущность Animal, агрегат AnimalAggregate
     2.2. Модель Enclosure: сущность Enclosure, агрегат EnclosureAggregate
     2.3. Модель Feeding: сущность FeedingSchedule, агрегат FeedingScheduleAggregate
  3. Слой представления также разбит на изолированные друг от друга контексты: AnimalFacadeContext, EnclosureFacadeContext, FeedingFacadeContext, StatisticsFacadeContext.
- **Единый язык**
  В моделях и методах используется язык предметной области. Примеры:
  1. Наименование доменной сущности “животное” - Animal
  2. Методы агрегата Animal - treat, declareSick, feed соответствуют понятиям предметной области - лечить, “признать заболевшим”, накормить.

Принципы Clean Architecture

- **Разделение на уровни**
  Реализованы слои Presentation,Application, Domain, Infrastructure в соответствующих Java-пакетах presentation, application, domain, infrastructure.
- **Граничные интерфейсы**
  Для взаимодействия с пользователем используется REST API и DTO.
- **Чистая зависимость**
  Зависимости между слоями реализованы через спецификацию. Например, в интерфейсе AnimalService из слоя Application реализована зависимость на слой

Infrastructure через интерфейс AnimalRepository и первый не видит детали реализации второго.