Отчёт о проекте зоопарка

Описание проекта

Веб-приложение разработано для автоматизации ключевых процессов Московского зоопарка: управление животными, вольерами, а также расписанием кормлений. Архитектура построена на принципах Clean Architecture и Domain-Driven Design (DDD). В проект интегрированы решения из предыдущего домашнего задания, включая расширенные характеристики животных (например, IQ, длина хвоста и пр.).

Основные возможности

Управление животными

- Добавление и удаление животных.
- Перемещение животных между вольерами с учетом их типа и совместимости.
- Просмотр полной информации о животных, включая дополнительные характеристики.

Управление вольерами

- Создание и удаление вольеров.
- Информация о типе, вместимости и текущем количестве животных.
- Функция уборки вольеров.

Управление расписанием кормления

- Просмотр общего и индивидуального расписания кормления.
- Добавление новых расписаний.
- Изменение времени кормления.
- Отметка выполнения кормления.

Статистика зоопарка

- Общее количество животных.
- Количество свободных вольеров.
- Статистика по типам животных и заполненности вольеров.

Архитектура

Принципы Clean Architecture

Domain

- Основные сущности: Animal, Enclosure, FeedingSchedule.

- Доменные события: AnimalMovedEvent, FeedingTimeEvent.
- Value Objects: AnimalStatus, AnimalType, EnclosureType, FoodType, Gender
- Изолированы от внешних слоев приложения.

Application

- Сервисы: AnimalTransferService, FeedingOrganizationService, ZooStatisticsService.
- Интерфейсы: IAnimalRepository, IEnclosureRepository, IFeedingScheduleRepository
- Используются интерфейсы для взаимодействия с хранилищами.

Infrastructure

- Хранилища в памяти: InMemoryAnimalRepository, InMemoryEnclosureRepository, InMemoryFeedingScheduleRepository.
- Эмуляция взаимодействия с внешними системами.

Presentation

- REST API контроллеры: AnimalController, EnclosureController, FeedingScheduleController, StatisticsController, TransferController.
- Принимают HTTP-запросы и возвращают ответы.

Применение Domain-Driven Design

Value Objects

- FoodType: тип пищи.
- AnimalStatus: состояние животного.
- AnimalType: тип животного
- EnclosureType: тип вольера
- Gender: гендер

Бизнес-логика инкапсулирована в доменных моделях

- Animal: методы Feed, Heal, MoveTo.
- Enclosure: AddAnimal, RemoveAnimal.
- FeedingSchedule: UpdateSchedule, MarkCompleted.

Доменные события

- AnimalMovedEvent — при перемещении животного.
- FeedingTimeEvent — при наступлении времени кормления.

Тестирование

Покрытие тестами: более 65% кода.

Контроллеры

- AnimalController
- EnclosureController

- FeedingScheduleController
- StatisticsController
- TransferController

Доменные объекты

- Animal
- Enclosure
- FeedingSchedule

Примеры тестов

- AnimalTests
- EnclosureTests

Соответствие требованиям

Требование	Реализация	Местоположение
Добавить/удалить животное	☑ Реализовано	AnimalController
Добавить/удалить вольер	☑ Реализовано	EnclosureController
Переместить животное между вольерами	☑ Реализовано	AnimalController, AnimalTransferService
Просмотреть расписание кормления	☑ Реализовано	FeedingScheduleController
Добавить новое кормление	☑ Реализовано	FeedingScheduleController
Просмотреть статистику	☑ Реализовано	StatisticsController, ZooStatisticsService

Вывод

☑ Приложение полностью соответствует требованиям:
- Все основные функции реализованы.
- Соблюдены принципы Clean Architecture и DDD.
- Код покрыт тестами более чем на 65%.