

Отчёт Мдз 2

Мушкат Илья Валерьевич БПИ238

1. Общее описание проекта

Проект является REST API-сервисом для управления зоопарком. Система должна позволять управлять животными и вольерами, фиксировать и изменять расписание кормлений, а также выполнять операции кормления и перемещения животных. Архитектура проекта реализована согласно принципам Clean Architecture и Domain-Driven Design (DDD), бизнес-логики изолирована и проект имеет строгую модульную структуру.

2. Архитектура проекта

Проект построен на основе Clean Architecture и включает в себя следующие слои:

- Domain: доменные сущности, бизнес-правила и интерфейсы.
- Application: сценарии использования и сервисы бизнес-логики.
- Infrastructure: реализация интерфейсов репозитория, in-memory хранилища.
- Presentation: REST API контроллеры.

3. Концепции Domain-Driven Design

- Использование Value Objects:
 - AnimalName, Food, Species как обёртки над примитивами, обеспечивают валидацию и целостность данных.
- Инкапсуляция бизнес-правил:
 - В классе Animal реализованы правила перемещения и кормления.
- Агрегаты и сущности:
 - Animal и Enclosure являются агрегатами, а FeedingSchedule управляет кормлением животных.

4. Принципы Clean Architecture

- Изоляция слоёв:
 - Слой Domain не зависит ни от одного из слоев.
- Зависимости только через интерфейсы:
 - Используются интерфейсы IAnimalRepository,

IFeedingScheduleRepository и IEnclosureRepository.

- Изолированная бизнес-логика:
 - Весь код, отвечающий за правила размещения, кормления и перемещения, расположен в Domain и Application.

5. Реализация Web API

В Presentation реализованы контроллеры:

- AnimalController:
 - GET /api/Animals — список животных
 - POST /api/Animals — добавление нового
 - DELETE /api/Animals/{id} — удаление
 - POST /api/Animals/{animalId}/transfer/{enclosureId} — перемещение
- EnclosureController:
 - GET /api/Enclosures
 - POST /api/Enclosures
 - DELETE /api/Enclosures/{id}
- FeedingScheduleController:
 - GET /api/FeedingSchedules
 - POST /api/FeedingSchedules
- ZooController:
 - GET /api/Statistics – получение всей информации о зоопарке

6. Тестирование через Swagger

Через Swagger UI были выполнены следующие сценарии:

- Добавление животных и вольеров
- Получение данных о животных и вольерах
- Создание расписания кормления
- Кормление животного
- Перемещение животного между вольерами

7. In-Memory хранилище

Хранилище данных реализовано в оперативной памяти с использованием коллекций. Классы `InMemoryAnimalRepository`, `InMemoryEnclosureRepository` и `InMemoryFeedingScheduleRepository` реализуют соответствующие интерфейсы, обеспечивая хранение и управление данными без зависимости от внешней базы данных.

8. Отражение пунктов функционала

Функционал	Класс / Модуль
Добавление/удаление животных	<code>AnimalController</code> , <code>AnimalService</code>
Перемещение животных	<code>AnimalService</code>
Работа с вольерами	<code>EnclosureController</code> , <code>EnclosureService</code>
Создание расписания кормления	<code>FeedingScheduleController</code> , <code>FeedingService</code>
In-Memory хранилище	<code>InMemoryAnimalRepository</code> , <code>InMemoryEnclosureRepository</code>
Тестирование через Swagger	Swagger UI

9. Пример работы

1) Добавляем животных без id и enclosureId

```
{
  "name": {
    "value": "Simba"
  },
  "species": {
    "value": "Lion"
  }
}

{
  "name": {
    "value": "Nikolaich"
  },
  "species": {
    "value": "Monkey"
  }
}

{
  "name": {
    "value": "Bobik"
  },
  "species": {
    "value": "Dog"
  }
}
```

2) Проверяем актуальный список животных

```
[
  {
    "id": "0f3ea6fd-1f5f-4177-a34f-ba602bd2e281",
    "name": {
      "value": "Bobik"
    },
    "species": {
      "value": "Dog"
    },
    "enclosureId": "00000000-0000-0000-0000-000000000000"
  },
  {
    "id": "c357f6a9-f8f5-4d33-b51f-54c030006fb6",
    "name": {
      "value": "Nikolaich"
    },
    "species": {
      "value": "Monkey"
    },
    "enclosureId": "00000000-0000-0000-0000-000000000000"
  },
  {
    "id": "3a4ba52a-6df3-41be-afce-f6f1ac627b41",
    "name": {
      "value": "Simba"
    },
    "species": {
      "value": "Lion"
    }
  }
]
```

3) Добавляем вольер

```
{
  "name": "Jungles",
  "type": "Jungle forest"
}
```

4) Получаем список вольеров и запоминаем id

```
[
  {
    "id": "f945b175-fc2b-40f5-b70c-81e26dd13791",
    "name": "Jungles",
    "type": "Jungle forest"
  }
]
```

5) Перемещаем нужных животных в вольер

```
{
  "animal": {
    "id": "c357f6a9-f8f5-4d33-b51f-54c030006fb6",
    "name": {
      "value": "Nikolaich"
    },
    "species": {
      "value": "Monkey"
    },
    "enclosureId": "f945b175-fc2b-40f5-b70c-81e26dd13791"
  },
  "fromEnclosureId": "00000000-0000-0000-0000-000000000000",
  "toEnclosureId": "f945b175-fc2b-40f5-b70c-81e26dd13791"
}
```

```
{
  "animal": {
    "id": "3a4ba52a-6df3-41be-afce-f6f1ac627b41",
    "name": {
      "value": "Simba"
    },
    "species": {
      "value": "Lion"
    },
    "enclosureId": "f945b175-fc2b-40f5-b70c-81e26dd13791"
  },
  "fromEnclosureId": "00000000-0000-0000-0000-000000000000",
  "toEnclosureId": "f945b175-fc2b-40f5-b70c-81e26dd13791"
}
```

6) Проверяем актуальный список животных

```
"name": {
  "value": "Bobik"
},
"species": {
  "value": "Dog"
},
"enclosureId": "00000000-0000-0000-0000-000000000000"
},
{
  "id": "c357f6a9-f8f5-4d33-b51f-54c030006fb6",
  "name": {
    "value": "Nikolaich"
  },
  "species": {
    "value": "Monkey"
  },
  "enclosureId": "f945b175-fc2b-40f5-b70c-81e26dd13791"
},
{
  "id": "3a4ba52a-6df3-41be-afce-f6f1ac627b41",
  "name": {
    "value": "Simba"
  },
  "species": {
    "value": "Lion"
  },
  "enclosureId": "f945b175-fc2b-40f5-b70c-81e26dd13791"
}
```

7) Добавим кормление и проверим его наличие в списке

```
{
  "animalId": "0f3ea6fd-1f5f-4177-a34f-ba602bd2e281",
  "feedingTime": "2025-04-19T11:54:23.912Z",
  "food": {
    "value": "Meat"
  }
}
```

```
[
  {
    "id": "3cbcd04c-6c14-43b1-8b4b-6cc4a35abb40",
    "animalId": "0f3ea6fd-1f5f-4177-a34f-ba602bd2e281",
    "feedingTime": "2025-04-19T11:54:23.912Z",
    "food": {
      "value": "Meat"
    }
  }
]
```

8) Проверим общую статистику по зоопарку

```
{
  "totalAnimals": 3,
  "totalEnclosures": 1,
  "emptyEnclosures": 0
}
```