КПО дз1
Балдина Анастасия, БПИ235

1. Применение принципов SOLID
   S - Пример: интерфейс Editor для редактирования и изменения объектов Inventory.
   1) Каждая реализация данного интерфейса предназначена для конкретного типа объекта Inventory (например, для изменения логики редактирования объекта Monkey нужно изменить только поведение класса MonkeyEditor).
   2) Разделение функционала: реализация Editor предназначена для изменения состояния объекта Inventory пользователем, а сам объект Inventory содержит только логику хранения состояния.
   O - Класс Animal открыт для расширения, но закрыт для изменения (имеет закрытые свойства). Например, инвентарный номер, кличка животного, количество потребляемой пищи.
   L - Пример: интерфейс Clinic и его метод examine(), который принимает в качестве аргумента интерфейс Observable.
   I - Пример: класс Tiger реализует 3 интерфейса - Inventory (спецификация инвентарного средства), Alive (спецификация живого существа) и Observable (спецификация пациента клиники).
   D - В проекте в качестве связей между сервисами используются только абстракции. Например, сервис ZooApiImpl имеет зависимость на абстракции Registry и Clinic, а их реализации ZooRegistry и ZooClinic внедряются с помощью DI-Spring Framework.

2. Применение DI-контейнера
   В качестве DI-контейнера используется класс AnnotationConfigApplicationContext. Конфигурация контекста описана с помощью классов EditorsConfiguration, ServiceConfiguration и UIConfiguration. В качестве bean-definition используются методы данных классов, имеющие аннотацию @Bean.

3. Ввод-вывод в консоль:

```
Select option:
1 Accept animals
2 Print animals
3 Print friendly & healthy animals (for contact zoo)
4 Examine animal in clinic
5 Print food requirements
6 Accept thing
7 Print things
8 Search inventory
9 Edit inventory
10 Remove inventory
11 Accept employee
12 Print employees
13 Exit
[1 ... 13]:
```

```
[1 ... 13]: 2
==== list of animals ====
N:1, Monkey: Lena
N:3, Monkey: Dasha
N:4, Rabbit: White
N:5, Rabbit: Grey
N:6, Tiger: Shar Khan
N:7, Wolf: North
N:8, Wolf: South
=========================
[1 ... 13]: 5
==== food requirements ===
Employee: 3
Monkey: 5
Rabbit: 2
Wolf: 9
Tiger: 15
==========================
[1 ... 13]: 12
=== list of employee ===
N:10, Director: Korshynov Lev Ivanovich
N:11, Veterenarian: Medvedeva Elena Potapovna
N:12, Security: Volkov Lev Petrovich
========================
```

```
[1 ... 13]: 8
=== search inventory ===
Input inventory number: 11
N:11, Employee: Medvedeva Elena Potapovna
    position: Veterenarian
    food count: 1
========================
[1 ... 13]: 7
==== list of things ===
N:2, Computer: Notebook Lenovo M16 2025
N:9, Table: Computers table
========================
[1 ... 13]: 8
=== search inventory ===
Input inventory number: 7
N:7, Wolf(predator): North
    is alfa: yes
    food count: 5
    is healthy: yes
    examine at: 14.02.2025 02:27:37
========================
[1 ... 13]: 3
==== list of friendly & healthy animals ====
N:1, Monkey: Lena
N:4, Rabbit: White
============================================
```

```
[1 ... 13]: 1
=== accept animal ===
Select option:
1 Monkey
2 Rabbit
3 Tiger
4 Wolf
5 Exit
[1 ... 5]: 1
Input monkey nickname: Lesha
Input food count: 2
Is monkey friendly? (y/n): y
Is monkey smart? (y/n): y
Is animal healthy? (y/n): y
Animal accepted: N:13, Monkey(herbo): Lesha
    food count: 2
    is healthy: yes
    is friendly: yes
    is smart: yes
    examine at: 14.02.2025 23:24:22
=====================
```

4. Code style
   Используется Java Camel code style (и из-за этого у интерфейсов нет “I” в начале имени).

5. Юнит-тесты
   Использовались библиотеки JUnit5 и Mockito.