CSCI 4448 Object-Oriented Analysis and Design
HW/Project 2
Athena Parker
Katariya Edfors
Question 2-5 Diagrams
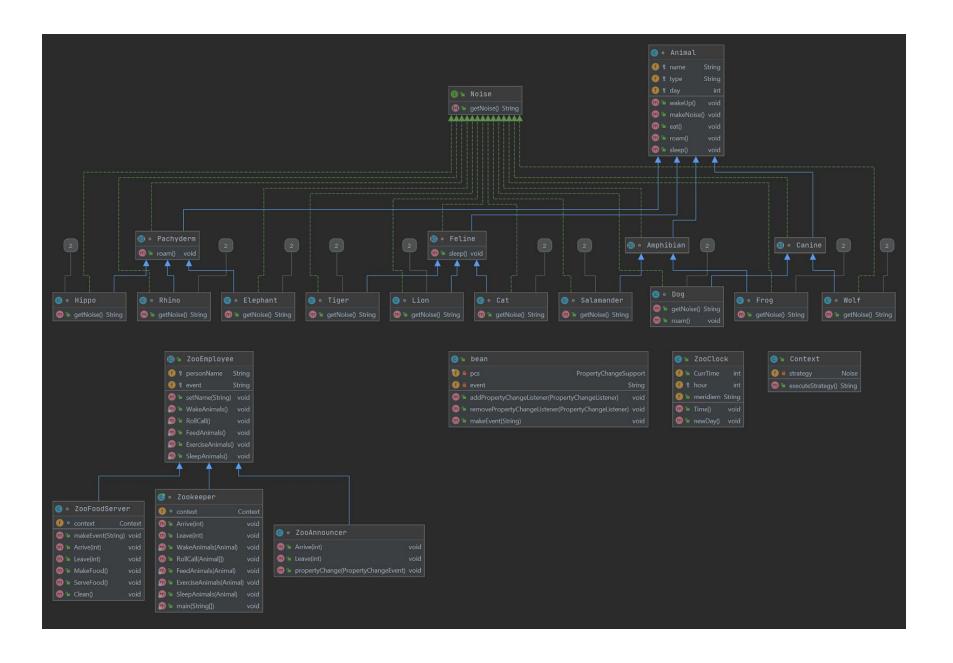
Contents:

Noise (interface)
- getNoise() String

Animal (class)
- name : String
- type : String
- day : int
- wakeUp() : void
- makeNoise() : void
- eat() : void
- roam() : void
- sleep() : void

Pachyderm
- roam() : void

Feline
- sleep() : void

Amphibian

Canine

Hippo
- getNoise() String

Rhino
- getNoise() String

Elephant
- getNoise() String

Tiger
- getNoise() String

Lion
- getNoise() String

Cat
- getNoise() String

Salamander
- getNoise() String

Dog
- getNoise() String
- roam() : void

Frog
- getNoise() String

Wolf
- getNoise() String

ZooEmployee
- personName : String
- event : String
- setName(String) : void
- WakeAnimals() : void
- RollCall() : void
- FeedAnimals() : void
- ExerciseAnimals() : void
- SleepAnimals() : void

bean
- pcs : PropertyChangeSupport
- event : String
- addPropertyChangeListener(PropertyChangeListener) : void
- removePropertyChangeListener(PropertyChangeListener) : void
- makeEvent(String) : void

ZooClock
- CurrTime : int
- hour : int
- meridiem : String
- Time() : void
- newDay() : void

Context
- strategy : Noise
- executeStrategy() : String

ZooFoodServer
- context : Context
- makeEvent(String) : void
- Arrive(int) : void
- Leave(int) : void
- MakeFood() : void
- ServeFood() : void
- Clean() : void

Zookeeper
- context : Context
- Arrive(int) : void
- Leave(int) : void
- WakeAnimals(Animal) : void
- RollCall(Animal[]) : void
- FeedAnimals(Animal) : void
- ExerciseAnimals(Animal) : void
- SleepAnimals(Animal) : void
- main(String[]) : void

ZooAnnouncer
- Arrive(int) : void
- Leave(int) : void
- propertyChange(PropertyChangeEvent) : void

Zookeeper

Arrive

Leave

Roll Call

Feed Animals

Exersige Animals

Sleep animals

Identify Zookeeper

Know each animal's noise

Identify Animals

<<includes>>

<<includes>>

<<extends>>

<<includes>>

<<includes>>

<<include>>

<<includes>>

Sequence diagram with participants: Actor, ZooKeeper, ZooAnnouncer, ZooFoodServer, ZooClock, Animal

- Actor → ZooKeeper: Input #of Days
- ZooKeeper → ZooClock: Iterate Day
- ZooClock ⇢ ZooKeeper: ZooClock.Time()

CurrTime is 8 am
- ZooKeeper → ZooAnnouncer: Announce Arrival
- ZooAnnouncer → ZooFoodServer: Announce Arrival

CurrTime = 9am
- ZooKeeper → ZooAnnouncer: make_event(wake)
  - Loop: for each animal in ZooAnimals
    - WakeAnimals → Animal

CurrTime 10am
- ZooKeeper → ZooAnnouncer: make)event(rollCall)
- RollCall(ZooAnimals) → Animal

12 pm
- makeEvent(Lunch is Served) ← ZooFoodServer

5pm
- makeEvent(Dinner is Served) ← ZooFoodServer

7pm
- ZooKeeper → ZooAnnouncer: MakeEvent(feed, exercise, sleep)
  - Loop: for each animal in ZooAnimals
    - FeedAnimal, ExerciseAnimal, SleepAnimal → Animal

8pm
- ZooKeeper → ZooAnnouncer: makeEvent(Leave)