**Prerequisite:**

Java 8+ and Apache Maven

Build project:

git clone https://github.com/SureshG02/zoo-application.git

Go to directory zoo-application and run below command:

mvn clean install

Run main class:

mvn exec:java -Dexec.mainClass="zoo.demo.App"

**About application:**

Zoo console application lets you manage below three options. Please enter options as number 1 or 2 or 3.

1 >> SHOW\_ANIMALS

Used for displaying all animals detail along with friends.

2 >> LIVE\_ONE\_DAY

Let’s each animal make friend or unfriend with other animal in zoo. Basically, works on below three rules.

* If A​ is a friend of B, then B ​is a friend of A ​automatically.
* Similarly, if A loses friendship with B then B automatically loses with A.
* Every day each animal randomly loses one friend (if there are any) and gets from zero to one new friend.

3 >> Exit

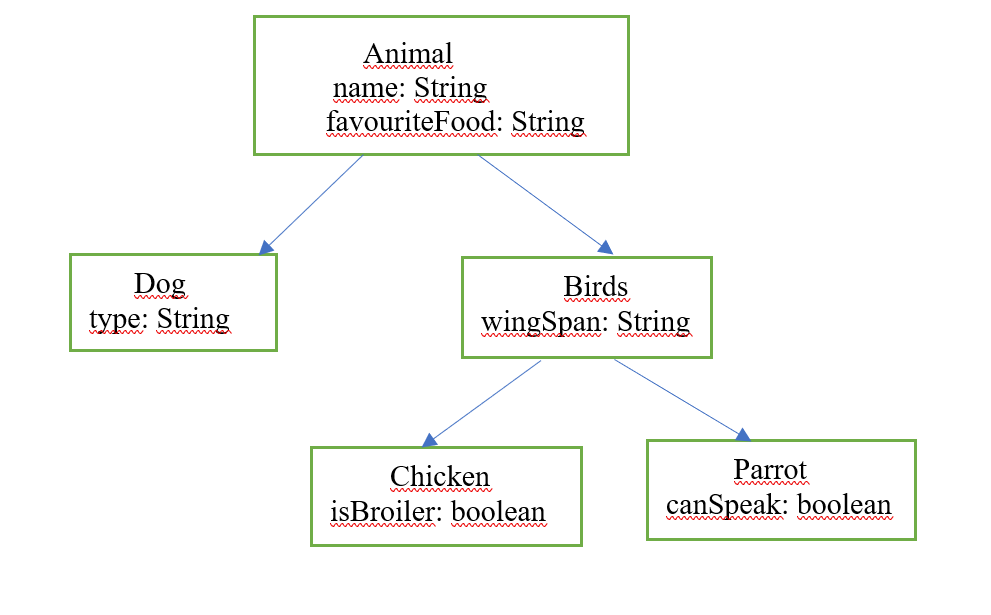
Exit console

**Technical details:**

Data structure friendMap<Animal, List<Animal>> is used to manage friendship between all animals in zoo.

|  |
| --- |
| k=dog\_1, v=friendList |
| k=dog\_2, v=friendList |
| k=dog\_3, v=friendList |
| k=parrot\_1, v=friendList |
| k=parrot\_2, v=friendList |
| k=chicken\_1, v=friendList |
| k=chicken\_2, v=friendList |

**Association between Java classes:**



**display\_AllAnimals ():** Iterate over friendMap and displays each animal details along with all friends.

**liveOneDay (List<Animal> animalList):** Let’s each animal make friend or unfriend with other animal in zoo.

**Algorithm for choosing random friend while breaking friendship:**

**random.nextInt(friends.size()):** is used to select random index from existing friends and calls method removeFriendsFromMap(animal\_1, animal\_2) to update friendList for both animals.

**Algorithm for choosing random friend while making new friends:**

**generateRandomIndex(int selfIndex, int removedIndex, List friends, List animalList)**: Method used to choose random index while adding friend to each animal in map. This method has below 2 rules.

**1.** If animal\_1 loses friendship with animal\_2, then generateRandomIndex method makes sure that return index is not of A(animal\_1) and B(animal\_2) and not from existing friendList instead some other animal in animalList.

**2.** generateRandomIndex returns index between 0 to 8. Note: When indexes are 7 or 8 then it means animal is not interested in making new friend. Index 0 to 6 means animal is ready to choose animal from animalList and make friend.

Once index is returned **addFriendsToMap(animal\_1, animalList.get(add));** method is called to establish friendship between both animals. Parameter “add” is random index returned by generateRandomIndex method.