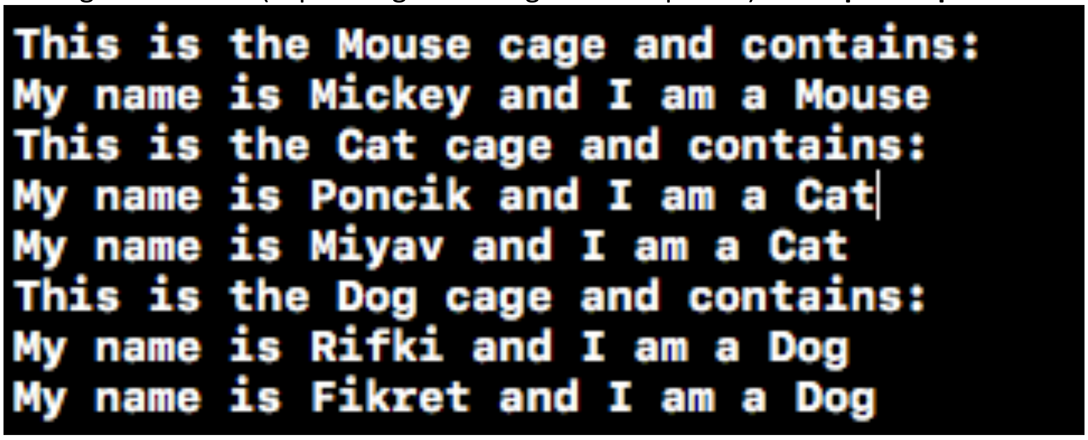
Week 3 - Practice Questions

1) A **queue** data structure is like stack data structure (we covered in class see Week3.playground if you need). Stack data structure is mainly referred as **LIFO** (Last in first out) since the last pushed item on to the stack is processed first when needed which we call pop. Push and pop are the main methods in a stack data structure. On the contrary, queue is mainly referred as **FIFO** (First in first out). The keywords are **enqueue** (like **push**) and **dequeue** (like **pop**). The both have some methods in common which are **isEmpty** in order to check if the stack or queue is empty. Stacks and queues can implemented using any data structure you want. E.g. arrays, linked lists and etc. Commonly arrays and linked lists are used which work in **O(1)** time. Giving this little information your task is to implement a **generic** queue data structure (which can take any type as input) using **"struct"** and as its container an **array** in swift. ***In addition you can give a constraint to your array such that it will have a limited amount of values that the queue can hold** (nothing has infinite space :D).

2) A zoo keeper cannot separate his animals and if you don't help him the animals will kill themselves. This zoo is capable of holding 3 species: dogs, cats and mice. First of all, you have to **enumerate** these animals conforming to the **String protocol**. Secondly, implement an **animal protocol** such that it has a variable **specie** of type **your enumeration name** and a **function** that describes the animal. After that, create an **animal class** conforming to the **animal protocol** you implemented before. The **animal class** should have the animal's name in addition as a property. Initialize it with the **specie** and **name**. Make as many species of animals you want. It is up to you. After having all these steps done, lastly, you have to create a **zoo class** such that it can hold the **animals** you created and **cages** in order to separate them. In the initialization phase initialize your containers for **animals** and **cages.** So, the zoo class should have 3 additional functions capable of adding animals, showing animals and sorting the animals (separating according to their species). **Example output:**



```
This is the Mouse cage and contains:
My name is Mickey and I am a Mouse
This is the Cat cage and contains:
My name is Poncik and I am a Cat
My name is Miyav and I am a Cat
This is the Dog cage and contains:
My name is Rifki and I am a Dog
My name is Fikret and I am a Dog
```

3) Last but not least, you mission is to turn the given code in the folder Week 3 named ***question3.playground*** to handle the errors using Error Handling we covered in class. HINT: Conform the Error protocol while enumerating and augment it with a function returning the error. Good Luck!