# Zoo

## Preparation

Download the skeleton provided in Judge. **Do not** change the **StartUp** class or its **namespace**.

## Problem description

*Create software that keeps track of the animals in a zoo.*

### Animal

You are given a class **Animal,** create the following fields:

* **Species – string**
* **Diet – string**
* **Weight – double**
* **Length – double**

The class **constructor** should receive **species, diet, weight, and length**.

Override **ToString()** method: **"The {animal specie} is a {diet} and weighs {weight} kg."**

### Zoo

Next, a class named **Zoo** is given that has a **collection**(**animals**) of type **Animal**. The name of the collection should be **Animals, which could not be modified**. All the entities of the **Animal** collection have the **same** properties. The **Zoo** also has some additional properties:

* **Name: string**
* **Capacity: int**

The **constructor** of the **Zoo** class should receive the **name and capacity**.

Implement the following features:

* string AddAnimal(Animal animal) – **adds** an **Animal** to the animals' collection **if** **there** **is** **room** for it. Before adding an animal, check:
  + - If the **animal** species is **null or whitespace**, return **"Invalid animal species."**
    - If the animal’s diet is different from **"herbivore"** or **"carnivore"**, return **"Invalid animal diet."**
    - If the **zoo is full** (there is no room for more animals), return **"The zoo is full."**
    - Otherwise, return: **"Successfully added {animal species} to the zoo."**
* int RemoveAnimals(string species) – removes all animals by **given species,** as a result, returns the **count** of the animals that were removed**.**
* List<Animal> GetAnimalsByDiet(string diet) – search and return a **list of animals** by **given diet.**
* Animal GetAnimalByWeight(double weight) – return the first animal, with the given weight**.**
* **string GetAnimalCountByLength(double minimumLength, double maximumLength)** –search of **all animals** which has a length between the given **(inclusively)**. As a result, return the following format: "**There are {count} animals with a length between {minimum length} and {maximum length} meters."**

## Constraints

* You will always have an animal added before receiving methods that manipulate the **animals in the Zoo**.

## Examples

This is an example of how the **Zoo class** is **intended to be used**.

|  |
| --- |
| Sample code usage |
| var zoo = new Zoo("Zoo Time", 20);  var animaleOne = new Animal("elephant", "herbivore", 4000, 4);  var animalTwo = new Animal("elephant", "herbivore", 3421.25, 3.7);  var animalThree = new Animal("zebra", "herbivore", 380.52, 1.9);  var animalFour = new Animal("cheetah", "carnivore", 59.52, 1.4);  var animalFive = new Animal("wolf", "carnivore", 65.25, 1.5);  Console.WriteLine(zoo.AddAnimal(animaleOne)); // Successfully added elephant to the zoo.  Console.WriteLine(zoo.AddAnimal(animalTwo)); // Successfully added elephant to the zoo.  Console.WriteLine(zoo.AddAnimal(animalThree)); // Successfully added zebra to the zoo.  Console.WriteLine(zoo.AddAnimal(animalFour)); // Successfully added cheetah to the zoo.  Console.WriteLine(zoo.AddAnimal(animalFive)); // Successfully added wolf to the zoo.  var animalByDiet = zoo.GetAnimalsByDiet("herbivore");  foreach (var animal in animalByDiet)  {  Console.WriteLine(animal.ToString());  }  // The elephant is a herbivore and weighs 4000 kg.  // The elephant is a herbivore and weighs 3421.25 kg.  // The zebra is a herbivore and weighs 380.52 kg.  var getAnimalByWeight = zoo.GetAnimalByWeight(4000);  Console.WriteLine(getAnimalByWeight.ToString());  // The elephant is a herbivore and weighs 4000 kg.  var animalSix = new Animal("wolf", "carnivore", 80.25, 1.8);  var animalSeven = new Animal("moose", "stake", 250.25, 2.5);  Console.WriteLine(zoo.AddAnimal(animalSix)); // Successfully added wolf to the zoo.  Console.WriteLine(zoo.AddAnimal(animalSeven)); // Invalid animal diet.  Console.WriteLine(zoo.GetAnimalCountByLength(1.4, 3));  // There are 4 animals with a length between 1.4 and 3 meters.  Console.WriteLine($"Animals living in the zoo: {zoo.Animals.Count}.");  // Animals living in the zoo: 6.  Console.WriteLine(zoo.RemoveAnimals("elephant")); // 2  Console.WriteLine($"There are {zoo.Animals.Count} animals living in the zoo.");  // Animals living in the zoo: 4. |

## Submission

Zip all the files in the project folder except the **bin** and **obj** folders.