# 03. Animal Shelter



## Preparation

Download the skeleton provided in Judge. **Do not** change the **packages**!

**Pay attention to name the package shelter, all the classes, their fields, and methods the same way they are presented in the following document. It is also important to keep the project structure as described.**

## Problem Description

Your task is to create a repository, which stores items by creating the classes described below.

First, write a Java class **Animal** with the following fields:

* **name: String**
* **age: int**
* **caretaker: String**

The class **constructor** should receive a **name, age** and **caretaker.** You need to create the appropriate **getters and setters**. The class should override the **toString()** method in the following format:

**"{name} {age} ({caretaker})"**

**Next**, write a Java class **Shelter** that has **data** (a List, which stores the Animals). All entities inside the repository have the **same fields**. Also, the **Shelter** class should have those fields:

* **capacity: int**

The class **constructor** should receive **capacity**, also it should initialize the **data** with a new instance of the collection**.** Implement the following features:

* Field **data** – **List** that holds added animals
* Method add(Animal animal) – **adds** an **entity** to the data **if** **there** **is** an **empty cell** for the animal.
* Method remove(String name) – removes the animal by **given name,** if such **exists**, and **returns boolean**.
* Method **getAnimal(String name, String caretaker)** – returns the animal with the **given name** and **caretaker** or **null if no such animal exists**.
* Method getOldestAnimal() – returns the oldest Animal.
* Getter getCount – **returns** the **number** of animals.
* **getStatistics()** – **returns** a **String** in the following **format**:
  + **"The shelter has the following animals:  
    {name} {caretaker}  
    {name} {caretaker}**

**(…)**"

## Constraints

* The **combinations** of **names** and **caretakers** will **always be unique**.
* The **age** of the animals will always be **positive**.

## Examples

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

|  |
| --- |
| Sample code usage |
| // Initialize the repository  Shelter shelter = new Shelter(15);  // Initialize entity  Animal animal = new Animal("Rex", 7, "Sara");  // Print Animal  System.out.println(animal); // Rex 7 (Sara)  // Add Animal  shelter.add(animal);  // Remove Animal  System.out.println(shelter.remove("Rex")); // true  System.out.println(shelter.remove("Cayra")); // false  Animal animal1 = new Animal("Bela", 3, "Sia");  Animal animal2 = new Animal("Stormy", 4, "George");  shelter.add(animal1);  shelter.add(animal2);  // Get Oldest Animal  Animal oldestAnimal = shelter.getOldestAnimal();  System.out.println(oldestAnimal); // Stormy 4 (George)  // Get Animal  Animal animal3 = shelter.getAnimal("Bela", "Sia");  System.out.println(animal3); // Bela 3 (Sia)  // Count  System.out.println(shelter.getCount()); // 2  // Get Statistics  System.out.println(shelter.getStatistics());  //The shelter has the following animals:  //Bela Sia  //Stormy George |

## Submission

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

Submit a **single .zip file**, containing **shelter package, with the classes inside (Animal, Shelter, and the Main class)**, there is no specific content required inside the Main class e. g. you can do any kind of local testing of your program there. However, there should be a **main(String[] args)** method inside.