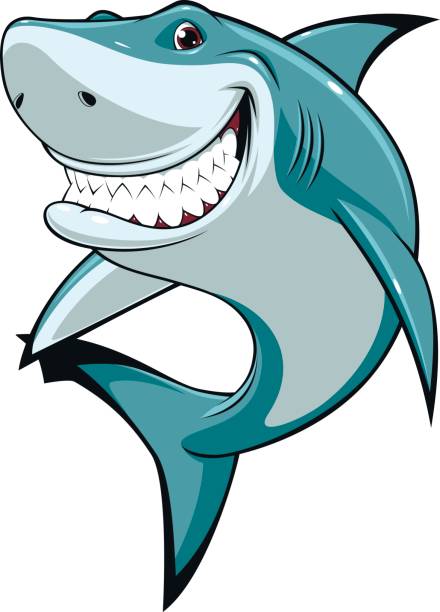
# 03. Chasing Sharks



***Sharks have been around for at least 420 million years. They are the top predators in the oceans. Let's find out why.***

## Preparation

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

**Pay attention to name the package sharkHaunt, 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 that stores sharks by creating the classes described below.

### Shark

First, write a class **Shark** with the following properties:

* **kind: String**
* **length: int**
* **food: String**
* **habitation: String**

The class **constructor** should receive **kind, length, food and habitation**. You need to create the appropriate **getters and setters**. All shark kinds will be **unique.** It is guaranteed that there **will be no duplicates** of kinds.

Override the **toString()** method in the following format:  
**The {kind} is {length} centimeters long, eats {food} and inhabits {habitation}.**

### Basin

**Next**, write a class **Basin** that has **data** (a List which stores the entity **Shark**). All entities inside the repository have the **same properties**. The **Basin** class should have those **properties**:

* **name: String**
* **capacity: int**
* **sharks: List<Shark>**

The class **constructor** should receive **name** and **capacity.** Also, it should initialise the **sharks** with a new **collection** instance.Implement the following features:

* **Method addShark(Shark shark)** – **adds** an **entity** to the data **if** a spacefor it, otherwise print: **"This basin is at full capacity!"**
* **Method removeShark(String kind)** – removes a shark by **given kind,** if such **exists**, and **returns boolean** (true if it is removed, otherwise – false)
* **Method getLargestShark()**– **returns** the **largest shark** by **length** in the given basin
* **Method getShark(String kind)** – **returns** the **shark** with the **given kind,** otherwise – returns **null**
* **Method getCount** – **returns** the **count** of **sharks** in the given basin
* **Method getAverageLength –** **returns** integer - the average length of **the sharks** in the given basin
* **Method report()** – **returns** a **string** in the following **format** (print the sharks in **order of addition**):
  + **"Sharks in {basin name}:  
    The {kind} is {length} centimeters long, eats {food} and inhabits {habitation}.  
    The {kind} is {length} centimeters long, eats {food} and inhabits {habitation}.  
     (…)"**

## Constraints

* The **kind** and **length** of the shark will always be **unique**.
* You will always have a shark added before receiving methods manipulating the Basins's shark.

## Examples

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

|  |
| --- |
| **Sample code usage** |
| //Initialize the repositories (Basin)  Basin pacific = **new** Basin("Pacific Ocean", 6);  Basin atlantic = **new** Basin("Atlantic Ocean", 2);  Basin ganges = **new** Basin("Ganges River", 1);  //Initialize entities (Shark)  Shark tigerShark = **new** Shark("Tiger shark", 300, "mammals", "saltwater");  Shark whaleShark = **new** Shark("Whale shark", 1200, "zooplankton", "saltwater");  Shark dwarfShark = **new** Shark("Dwarf lantern shark", 20, "shrimp", "saltwater");  Shark bullShark = **new** Shark("Bull shark", 330, "dolphins", "freshtwater");  Shark gangesShark = **new** Shark("Ganges shark", 178, "fish", "freshwater");  //Add Shark  pacific.addShark(tigerShark);  pacific.addShark(whaleShark);  pacific.addShark(dwarfShark);  atlantic.addShark(bullShark);  ganges.addShark(gangesShark);  //Remove Shark  System.***out***.println(pacific.removeShark("Carpet shark")); //false  System.***out***.println(atlantic.removeShark("Lemon shark")); //false  System.***out***.println(atlantic.removeShark("Bull shark")); //true  //Get the largest shark  System.***out***.println(pacific.getLargestShark().getKind()); //Whale shark  //Get the average length of sharks in given basin  System.***out***.println(pacific.getAverageLength()); //506  System.***out***.println(atlantic.getCount()); //0  //Get a report for the given basin  System.***out***.println(pacific.report());  System.***out***.println(ganges.report());  //Sharks in Pacific Ocean:  //The Tiger shark is 300 centimeters long, eats mammals and inhabits saltwater.  //The Whale shark is 1200 centimeters long, eats zooplankton and inhabits saltwater.  //The Dwarf lantern shark is 20 centimeters long, eats shrimp and inhabits saltwater.  //Sharks in Ganges River:  //The Ganges shark is 178 centimeters long, eats fish and inhabits freshwater. |

## Submission

Submit **single .zip file**, containing **sharkHaunt** package, **with the classes inside** (**Basin** and **Shark** 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 **main(String[] args)** method inside.