**Fishing Net**

* **Preparation**

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

* **Problem description**

*Peter loves spinning fishing, and he needs a portable fishing net, where he can store his catch.*

**Fish**

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

* **FishType: string**
* **Length: double**
* **Weight: double**

The class **constructor** should receive **fish type, length, and weight**.

The class should also have a method:

* Override the **ToString()** method in the format: **"There is a {fishType}, {length} cm. long, and {weight} gr. in weight."**

**Net**

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

* **Material: string**
* **Capacity: int**

The **constructor** of the **Net** class should receive the **material, and capacity**.

Implement the following features:

* Getter **Count** - returns the **total count** of the fish in the net.
* **string AddFish(Fish fish)** - **adds** a **Fish** to the fish's collection **if** **there** **is** **room** for it. Before adding a fish, check:
* If the **fish type** is **null or whitespace**.
* If the fish’s **length** or **weight** is zero or less.

If the **fish type, length,** or **weight** properties are not valid**"**, return: **Invalid fish."**. If the **net is full** (there is no room for more fish), return **"Fishing net is full.".** Otherwise, return: **"Successfully added {fishType} to the fishing net."**

* **bool ReleaseFish(double weight)** – removes a fish by **given weight,** if such **exists return true**, otherwise **false.**
* **Fish GetFish(string fishType)** – search and returns a fish by **given fish type.**
* **Fish GetBiggestFish()**– search and returns the **longest fish** in the collection**.**
* **Report()** -returns information about the **Net** and all fish that were **not released, order by fish's length descending** in the following format:
* **"Into the** **{material}:  
  {Fish1}  
  {Fish2}  
  (…)**"

**Note: Do not use** "\n\r" **for a new line.**

* **Constraints**
* You will always have a fish added before receiving methods that manipulate the **fish in the Net**.
* **Examples**

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

|  |
| --- |
| **Sample code usage** |
| // Initialize the repository (Net)  Net net = new Net("cast net", 10);  // Initialize entity  Fish fishOne = new Fish("salmon", 44.25, 941.15);  Fish fishTwo = new Fish("catfish", 105.25, 2100.15);  Fish fishThree = new Fish("bass", 25.25, 321.15);  // Add Fish  Console.WriteLine(net.AddFish(fishOne)); // Successfully added salmon to the fishing net.  Console.WriteLine(net.AddFish(fishTwo)); // Successfully added carfish to the fishing net.  Console.WriteLine(net.AddFish(fishThree));// Successfully added bass to the fishing net.  Console.WriteLine(net.Count); // 3  foreach (var fish in net.Fish)  {  Console.WriteLine(fish.ToString());  // There is a salmon, 44.25 cm. long, and 941.15 gr. in weight.  // There is a carfish, 105.25 cm. long, and 2100.15 gr. in weight.  // There is a bass, 25.25 cm. long, and 321.15 gr. in weight.  }  // Remove Fish  Console.WriteLine(net.ReleaseFish(321.15)); // True  Console.WriteLine(net.Count); // 2  Fish fishFour = new Fish("mullet", 15.21, 150.33);  Fish fishFive = new Fish("barbel", 21.33, 190.14);  Fish fishSix = new Fish("trout", 38.35, 357.41);  // Add Fish  Console.WriteLine(net.AddFish(fishFour)); // Successfully added мullet to the fishing net.  Console.WriteLine(net.AddFish(fishFive)); // Successfully added barbel to the fishing net.  Console.WriteLine(net.AddFish(fishSix)); // Successfully added trout to the fishing net.  // GetFish  Console.WriteLine(net.GetFish("trout"));  // There is a trout, 38.35 cm. long, and 357.41 gr. in weight.  // GetBiggestFish  Console.WriteLine(net.GetBiggestFish());  // There is a carfish, 105.25 cm. long, and 2100.15 gr. in weight.  Console.WriteLine("----------------------Report----------------------");  Console.WriteLine(net.Report());  /\*  Into the cast net:  There is a catfish, 105.25 cm. long, and 2100.15 gr. in weight.  There is a salmon, 44.25 cm. long, and 941.15 gr. in weight.  There is a trout, 38.35 cm. long, and 357.41 gr. in weight.  There is a barbel, 21.33 cm. long, and 190.14 gr. in weight.  There is a mullet, 15.21 cm. long, and 150.33 gr. in weight.  \*/ |

* **Submission**

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