Lab: SOLID

Problems for in-class lab for the Python OOP Course @SoftUni.

1. Books

Refactor the provided code, so there is a separate class called **`Library`**, which contains **books** and has a method called **`find_book(title)`** that returns the book with the given **title**. **Remove** the **unnecessary** code from the **Book** class.

2. Animals

Refactor the provided code, so you **do not** need to make **any changes** in it when you want to **add** new species to the animals' list

3. Ducks

Refactor the provided code so it is in line with the **Liskov Substitution Principle**.

4. Entertainment System

We have been hired to create a game where the player sets up **entertainment systems**. Each piece of the system (television, game console, etc.) uses a specific **cable** to **connect** to another device. The **TV** uses an **HDMI** cable to connect to a game console. Both the **game console** and **TV connect** to a router via an **ethernet cable** to access the internet. And lastly, all the devices are connected to the wall via a **power cable** so they can turn on. Your job is to **extend** this behavior in the device classes.

5. Print books

We want to **print books**, but before printing the book, we should **format it**. To accomplish this, we have a class **Printer** that can print books and a **class Formatter** which is used by the Printer. **Refactor** the provided code that breaks the DIP because both Printer and Formatter **depend on concretions**, not abstractions, by **creating** abstractions and injecting them wherever needed.