Beqa Kopadze

**Final Exam**

**OOP-task6**

**Report Outline**

1. **Introduction**
2. **Class Descriptions**
   * ElectronicDevice
   * ElectronicsStore
   * LegalEntity
   * StoreTester
3. **Functionality and Goals**
4. **Conclusion**

**Report Content**

**1. Introduction**

The purpose of this report is to provide a comprehensive overview of the "Electronics Store" project, which is designed to manage a list of electronic devices in a store. This project consists of multiple classes that work together to add, remove, and manage electronic devices while adhering to certain legal entity requirements.

**2. Class Descriptions**

**ElectronicDevice**

* **Functionality**: This class represents an electronic device with attributes such as name, model, and serial number.
* **Goal**: To encapsulate the properties and behaviors of an electronic device.

**ElectronicsStore**

* **Functionality**: This class implements the LegalEntity interface and manages a list of ElectronicDevice instances. It provides methods to add, remove, and list devices, as well as get the count of devices in the store.
* **Goal**: To simulate an electronics store that maintains a catalog of electronic devices and performs operations on this catalog.

**LegalEntity**

* **Functionality**: This interface defines the basic legal attributes that a legal entity should have, such as address and VAT number.
* **Goal**: To ensure that any class representing a legal entity (such as ElectronicsStore) has these basic attributes.

**StoreTester**

* **Functionality**: This class contains the main method to test the functionality of the ElectronicsStore and its interaction with ElectronicDevice instances.
* **Goal**: To demonstrate and validate the functionalities provided by the ElectronicsStore class.

**3. Functionality and Goals**

**Adding Devices** The addDevice method in ElectronicsStore checks if a device already exists in the store (by serial number) before adding it. This prevents duplicate entries.

**Removing Devices** The deleteDevice method provides two ways to remove a device: by passing an ElectronicDevice object or by its serial number. This flexibility allows users to manage the store's inventory effectively.

**Listing Devices** The deviceList method prints all devices currently in the store, providing an overview of the store's inventory.

**Counting Devices** The deviceCount method returns and prints the number of devices in the store, giving a quick summary of the store's size.

**Implementing LegalEntity** The ElectronicsStore class implements the LegalEntity interface, ensuring it has an address and VAT number, which are essential for any legal entity operating within a regulatory framework.

**4. Conclusion**

The "Electronics Store" project successfully demonstrates the creation and management of an electronic devices store using object-oriented programming principles. By implementing the LegalEntity interface, it also ensures compliance with basic legal requirements for entities. The classes work together to provide a robust solution for managing an inventory of electronic devices