Ești dezvoltator software la o companie care produce drone comerciale. Lucrezi la software-ul pentru drone și trebuie să implementezi diferite module conform următoarelor cerințe:

1. Modulul de conexiune al dronei trebuie să permită altor dezvoltatori să se conecteze la dispozitiv ținând cont de faptul că trebuie să existe o conexiune unică. Conexiunea trebuie să implementeze interfața `IDroneConnection`. Implementează modulul `DroneConnection` astfel încât să nu fie posibilă crearea a mai mult de o conexiune.

2. În backend, trebuie să gestionezi conexiunea către diferite microservicii REST API, cum ar fi serviciul de utilizatori, serviciul de date meteo, serviciul cu date de fabricație, etc. Fiecare serviciu are un nume unic și o adresă URL asociată, bazate pe clasa `AbstractMicroService`. De asemenea, fiecare serviciu acceptă o singură conexiune de la fiecare client. Implementează o clasă care va gestiona conexiunile unice pentru microservicii, extinzând `AbstractMicroService`.

3. Drona poate fi accesorizată cu diferite module (vedere pe timp de noapte, GPS tracking, cameră de înaltă rezoluție, urmărire inteligentă, baterie suplimentară etc.). Framework-ul modulelor extinde clasa `AbstractModule`. Deoarece fiecare modul are atribute unice (2-3 atribute suplimentare față de cele din `AbstractModule`), iar acestea se pot schimba în viitor, implementează un strat intermediar care să permită altor dezvoltatori să creeze/utilizeze module diferite într-un mod simplificat și independent de tipul de modul.

4. Clasa `Drone` va gestiona toate detaliile dronei. Adaugă mai multe atribute (cel puțin 4-5) și implementează un mecanism care să permită altor dezvoltatori să creeze eficient o dronă cu orice combinație de module/atribute, fără posibilitatea de a le modifica ulterior (odată ce drona este creată, procesul de fabricație începe și nu se mai poate schimba).

5. Anumite modele de drone sunt solicitate mai frecvent decât altele. Pentru a facilita crearea acestor tipuri specifice, implementează un catalog de modele de drone predefinite (cel puțin 3 tipuri) care să permită altor dezvoltatori să le instanțieze rapid și simplu.

6. Lucrezi la un modul AI care va fi instalat pe drone pentru a le permite să fie autonome. Clasa care gestionează modulul AI se numește `AutonomousDriving`. În faza de testare, multe instanțe ale aceleiași versiuni de modul AI sunt create. Implementarea actuală necesită mult timp pentru instanțierea entităților AI. Implementează o soluție care să rezolve această problemă, permițând dezvoltatorilor să creeze rapid noi instanțe ale modulelor AI.

7. Deoarece există diferite versiuni (cel puțin 2 sau 3) ale modulului AI, implementează o soluție care să permită dezvoltatorilor să creeze rapid copii ale acestora.

```
public interface IDroneConnection {
    void connect();
    void disconnect();
    boolean isConnected();
}

public abstract class AbstractMicroService {
    protected String serviceName;
    protected String serviceUrl;

    public AbstractMicroService(String name, String url) {
        this.serviceName = name;
        this.serviceUrl = url;
    }

    public abstract void connect();
}

public abstract class AbstractModule {
     protected String name;
     protected String description;
     protected float price;
}
```

```
public class Drone {
	private String model;
	private String softwareVersion;
	private float maxSpeed;
	private AbstractModule extraBattery;
}

public class AutonomousDriving {
	private String version;

	public AutonomousDriving(String version) throws
InterruptedException {
		this.version = version;
		System.out.println("Start AI module....");
		Thread.sleep(5000);
		System.out.println("Init AI module....learning done");
	}
}
```