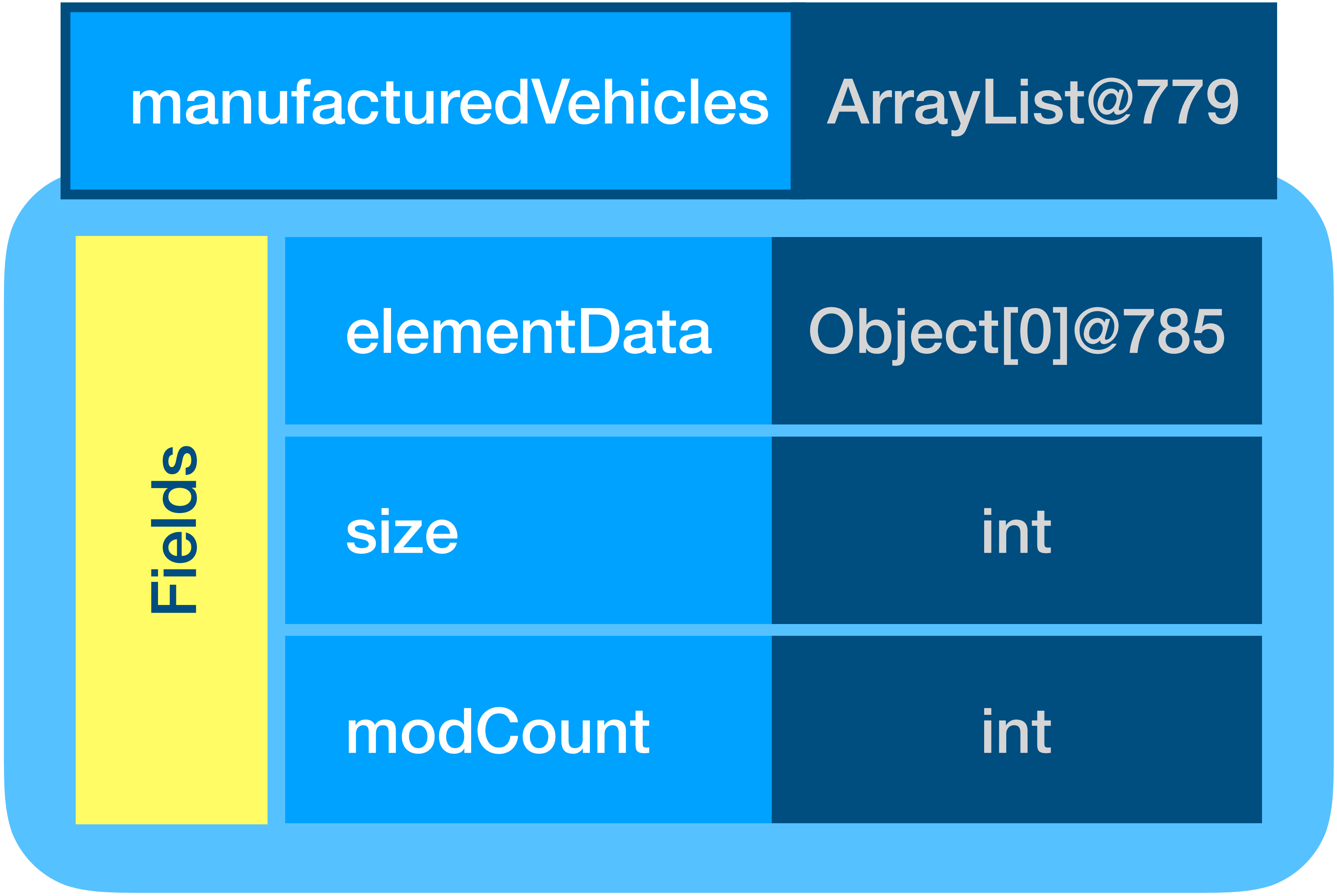

manufacturedVehicles ArrayList@779

Fields

elementData	Object[0]@785
size	int
modCount	int

Is that legal to have fields in interfaces?

No, since it’s a final or constant field you cannot modify it after the initialization! However, if you assign an object to that, you can modify the object as long as the reference is not changed.

Can I modify their values?

Yes, but only static final fields! (final by default)

What if we add more than 10 items to a final ArrayList?

Some initial questions would be…

```java
public interface Vehicle {

    static final ArrayList<String> manufacturedVehicles = new ArrayList<>();

}
```

```java
public class Car implements Vehicle {
    // Constructor method
    public Car(String carName) {
        Vehicle.manufacturedVehicles.add(carName);
    }
}
```

```java
public class Main {

    public static void main(String[] args) {

        Car audi = new Car("Audi");
        Car fiat = new Car("Fiat");
        Car mazda = new Car("Mazda");
        Car renault = new Car("Renault");
        Car toyota = new Car("Toyota");
        Car lotus = new Car("Lotus");
        Car benz = new Car("Benz");
        Car jaguar = new Car("Jaguar");
        Car kia = new Car("Kia");
        Car lexus = new Car("Lexus");

        Car volvo = new Car("Volvo");
        Car ford = new Car("Ford");
        Car bmw = new Car("BMW");

    }
}
```

A big picture of the hole source code