Graphical user interface, text

Description automatically generated

**Lesson 1 Demo 3**

**Create Chaincode for Agro Farm**

|  |
| --- |
| **Objective:** To create Chaincode for a Agro farm  **Tools required:** Ubuntu and Eclipse IDE  **Prerequisites:** Lesson 1 Demo 2 |

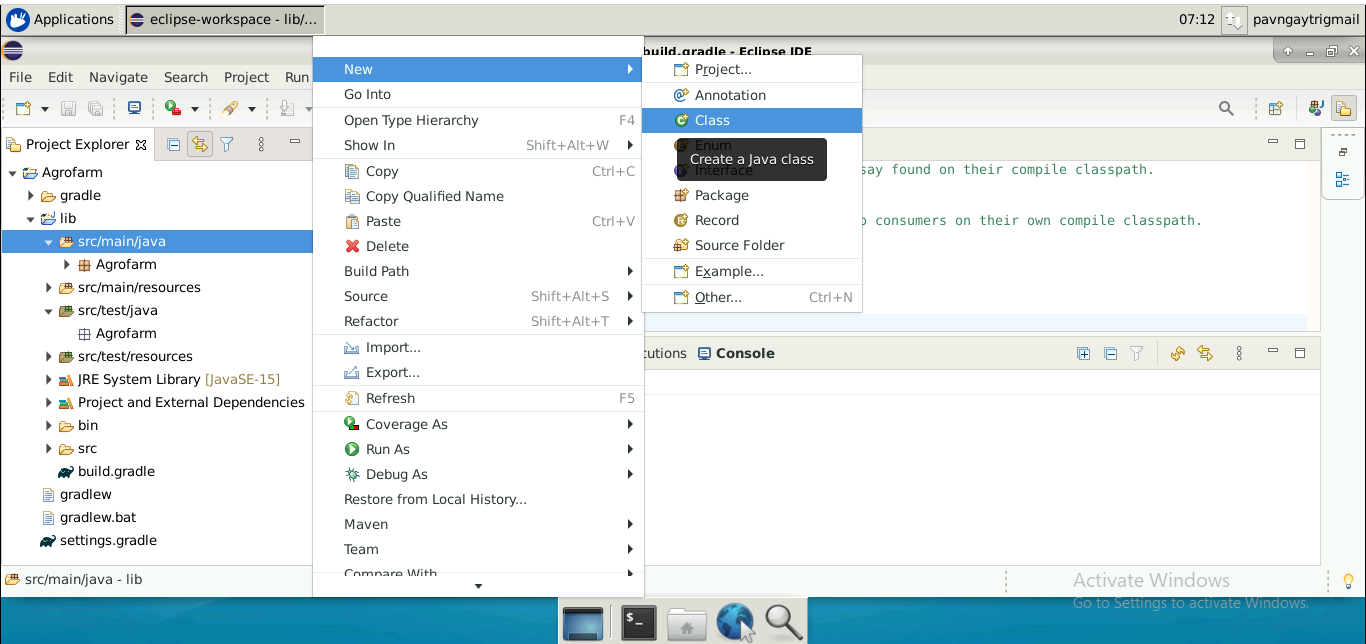
Steps to be followed:

1. Creating a Agroproduct.java file
2. Creating a AgroproductTransfer.java file
3. Compiling the chaincode in the Terminal

**Step 1: Creating a Agroproduct.java file**

1. Right-click on the project and navigate to **New** > **Class** to create a

**Agroproduct.java** class



1. Add the following code in the **Agroproduct.java** file:

***package Agrofarm;***

***import com.owlike.genson.annotation.JsonProperty;***

***import org.hyperledger.fabric.contract.annotation.DataType;***

***import org.hyperledger.fabric.contract.annotation.Property;***

***import java.util.Objects;***

***@DataType()***

***public final class Agroproduct {***

***@Property()***

***private final String id;***

***@Property()***

***private final String product;***

***@Property()***

***private final String producer;***

***@Property()***

***private final String produceraddress;***

***@Property()***

***private final String harvestdate;***

***@Property()***

***private final String priceperquintal;***

***public String getId() {***

***return id;***

***}***

***public String getProduct() {***

***return product;***

***}***

***public String getProducer() {***

***return producer;***

***}***

***public String getProducerAddress() {***

***return produceraddress;***

***}***

***public String getHarvestDate() {***

***return harvestdate;***

***}***

***public String getPricePerQuintal() {***

***return priceperquintal;***

***}***

***public Agroproduct(@JsonProperty("id") final String id, @JsonProperty("product") final String product, @JsonProperty("producer") final String producer,***

***@JsonProperty("produceraddress") final String produceraddress, @JsonProperty("harvestdate") final String harvestdate, @JsonProperty("priceperquintal") final String priceperquintal) {***

***this.id = id;***

***this.product = product;***

***this.producer = producer;***

***this.produceraddress = produceraddress;***

***this.harvestdate = harvestdate;***

***this.priceperquintal = priceperquintal;***

***}***

***@Override***

***public boolean equals(final Object obj) {***

***if (this == obj) {***

***return true;***

***}***

***if ((obj == null) || (getClass() != obj.getClass())) {***

***return false;***

***}***

***Agroproduct other = (Agroproduct) obj;***

***return Objects.deepEquals(new String[]{getId(), getProduct(), getProducer(), getProducerAddress(), getHarvestDate(), getPricePerQuintal()},***

***new String[]{other.getId(), other.getProduct(), other.getProducer(), other.getProducerAddress(), other.getHarvestDate(), other.getPricePerQuintal()});***

***}***

***@Override***

***public int hashCode() {***

***return Objects.hash(getId(), getProduct(), getProducer(), getProducerAddress(), getHarvestDate(), getPricePerQuintal());***

***}***

***@Override***

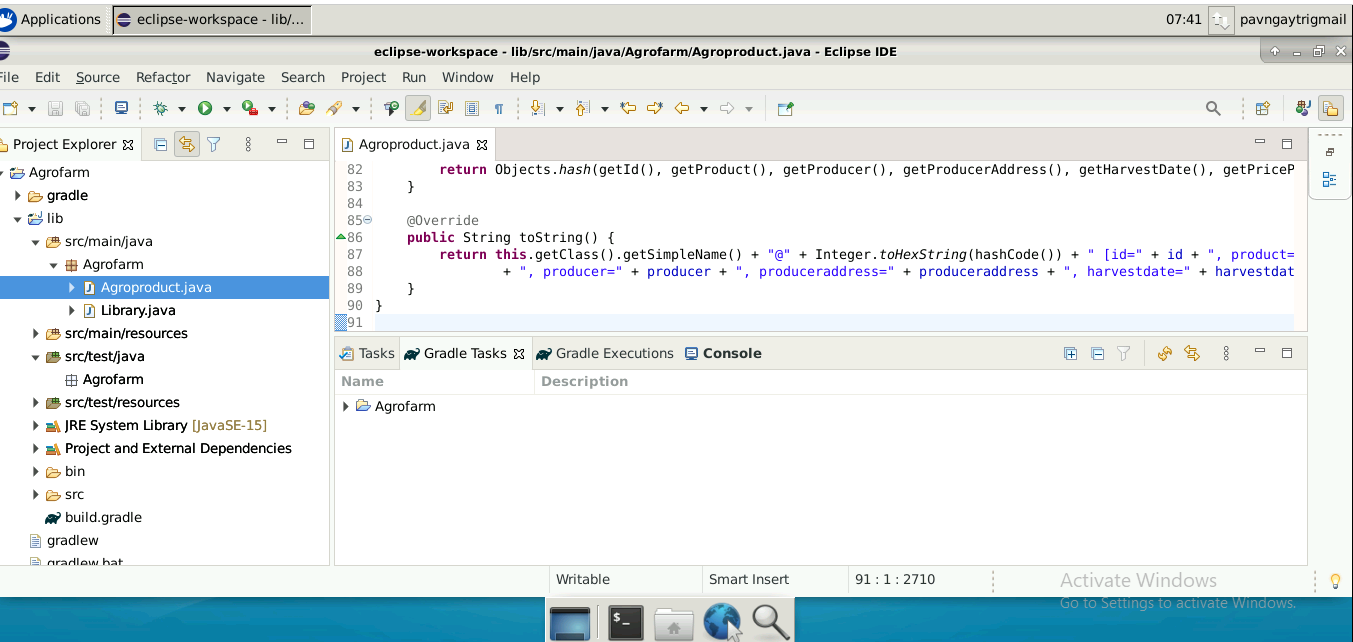
***public String toString() {***

***return this.getClass().getSimpleName() + "@" + Integer.toHexString(hashCode()) + " [id=" + id + ", product=" + product***

***+ ", producer=" + producer + ", produceraddress=" + produceraddress + ", harvestdate=" + harvestdate + ", priceperquintal=" + priceperquintal + "]";***

***}***

***}***



**Step 2: Creating a AgroproductTranfer.java file**

Follow Step 1.1 to create a **AgroproductTransfer.java** class and add the following code in it:

***package Agrofarm;***

***import org.hyperledger.fabric.contract.Context;***

***import org.hyperledger.fabric.contract.ContractInterface;***

***import org.hyperledger.fabric.contract.annotation.Contract;***

***import org.hyperledger.fabric.contract.annotation.Default;***

***import org.hyperledger.fabric.contract.annotation.Info;***

***import org.hyperledger.fabric.contract.annotation.Transaction;***

***import org.hyperledger.fabric.shim.ChaincodeException;***

***import org.hyperledger.fabric.shim.ChaincodeStub;***

***import com.owlike.genson.Genson;***

***@Contract(***

***name = "Agrofarm",***

***info = @Info(***

***title = "Agrofarm contract",***

***description = "A Sample Agroproduct transfer chaincode example",***

***version = "0.0.1-SNAPSHOT"***

***)***

***)***

***@Default***

***public final class AgroproductTransfer implements ContractInterface {***

***private final Genson genson = new Genson();***

***private enum AgrofarmErrors {***

***Agroproduct\_NOT\_FOUND,***

***Agroproduct\_ALREADY\_EXISTS***

***}***

***@Transaction()***

***public void initLedger(final Context ctx) {***

***ChaincodeStub stub = ctx.getStub();***

***Agroproduct Agroproduct = new Agroproduct("1", "Dates", "Mohan", "Rajasthan,India", "10-02-2024", "6000");***

***String AgroproductState = genson.serialize(Agroproduct);***

***stub.putStringState("1", AgroproductState);***

***}***

***@Transaction()***

***public Agroproduct addNewAgroproduct(final Context ctx, final String id, final String product,***

***final String producer, final String produceraddress,***

***final String harvestdate, final String price\_per\_quintal) {***

***ChaincodeStub stub = ctx.getStub();***

***String AgroproductState = stub.getStringState(id);***

***if (!AgroproductState.isEmpty()) {***

***String errorMessage = String.format("Agroproduct %s already exists", id);***

***System.out.println(errorMessage);***

***throw new ChaincodeException(errorMessage, AgrofarmErrors.Agroproduct\_ALREADY\_EXISTS.toString());***

***}***

***Agroproduct Agroproduct = new Agroproduct(id, product, producer, produceraddress, harvestdate, price\_per\_quintal);***

***AgroproductState = genson.serialize(Agroproduct);***

***stub.putStringState(id, AgroproductState);***

***return Agroproduct;***

***}***

***@Transaction()***

***public Agroproduct changeAgroproductOwnership(final Context ctx, final String id, final String newAgroproductProducer) {***

***ChaincodeStub stub = ctx.getStub();***

***String AgroproductState = stub.getStringState(id);***

***if (AgroproductState.isEmpty()) {***

***String errorMessage = String.format("Agroproduct %s does not exist", id);***

***System.out.println(errorMessage);***

***throw new ChaincodeException(errorMessage, AgrofarmErrors.Agroproduct\_NOT\_FOUND.toString());***

***}***

***Agroproduct Agroproduct = genson.deserialize(AgroproductState, Agroproduct.class);***

***Agroproduct newAgroproduct = new Agroproduct(Agroproduct.getId(), Agroproduct.getProduct(),***

***newAgroproductProducer, Agroproduct.getProducerAddress(), Agroproduct.getHarvestDate(),***

***Agroproduct.getPricePerQuintal());***

***String newAgroproductState = genson.serialize(newAgroproduct);***

***stub.putStringState(id, newAgroproductState);***

***return newAgroproduct;***

***}***

***@Transaction()***

***public Agroproduct queryAgroproductById(final Context ctx, final String id) {***

***ChaincodeStub stub = ctx.getStub();***

***String AgroproductState = stub.getStringState(id);***

***if (AgroproductState.isEmpty()) {***

***String errorMessage = String.format("Agroproduct %s does not exist", id);***

***System.out.println(errorMessage);***

***throw new ChaincodeException(errorMessage, AgrofarmErrors.Agroproduct\_NOT\_FOUND.toString());***

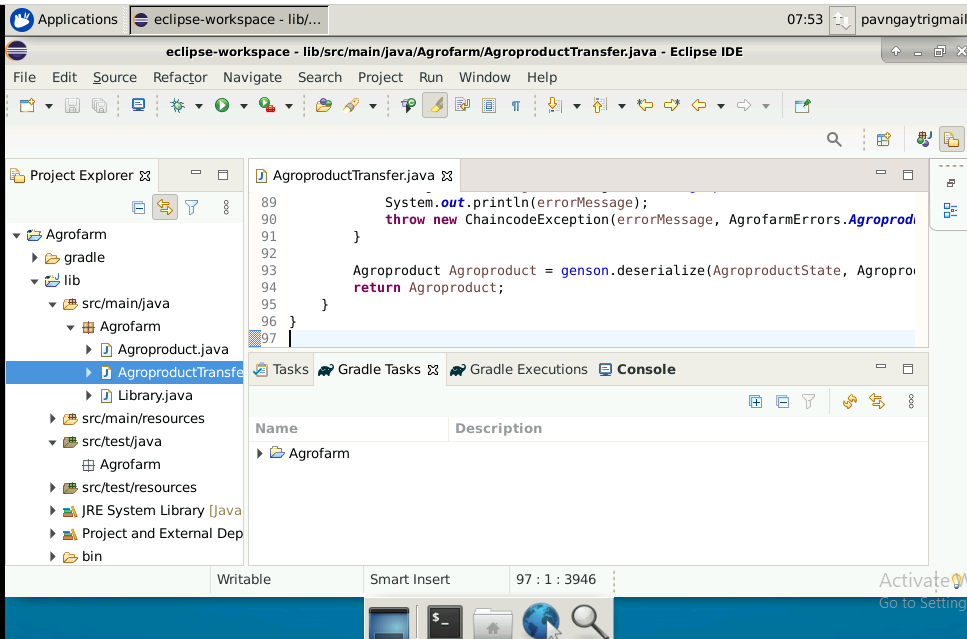
***}***

***Agroproduct Agroproduct = genson.deserialize(AgroproductState, Agroproduct.class);***

***return Agroproduct;***

***}***

***}***

****

**Step 3: Compiling the project**

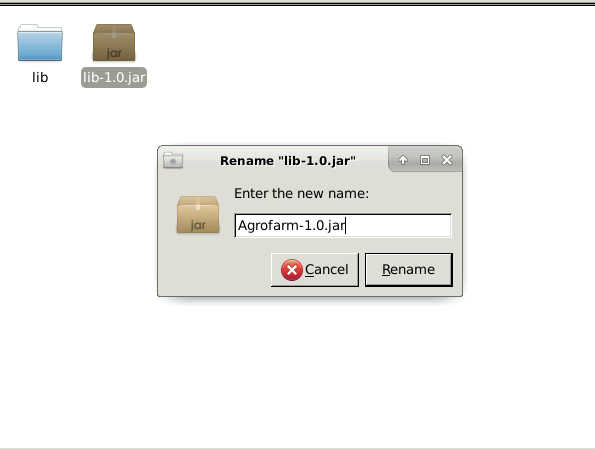
1. To compile the chaincode, run the command on the terminal:

**cd eclipse-workspace/Agrofarm/**

***./gradlew installDist***

1. Rename the compiled file. Go to eclipse-workspace/Agrofarm/lib/build/install/lib/ folder

Rename the **lib-1.0.jar** file to **Agrofarm-1.0.jar**

****

The chaincode for a Agro farm is successfully created.