package player;

import edu.neu.ccs.demeterf.demgen.lib.List;
import gen.*;

/**
  * Player interface to reveal the various agent implementations.
  */
public interface PlayerI {
  /**
   * Responsible for derivative creation.
   */
  interface CreateAgentI {
    /**
      * Returns a newly created derivative of a different type than already
      * existing derivatives.
      */
    Derivative createDerivative(Player player, List<Type> existing);
  }

  /**
   * Responsible for buying/selection.
   */
  interface BuyAgentI {
    /**
      * Returns a list of derivatives to be purchased.
      */
    List<Derivative> buyDerivatives(List<Derivative> forSale, double account);
  }

  /**
   * Responsible for raw material delivery.
   */
  interface DeliverAgentI {
    /**
      * Returns raw materials for the given derivatives.
      */
    Derivative deliverRawMaterial(Derivative needRM);
  }

  /**
   * Responsible for raw material finishing.
   */
  interface FinishAgentI {
    /**
      * Returns finished product for the given derivative.
      */
    FinishedProduct finishDerivative(Derivative needFinish);
  }

  /**
   * Returns the player’s simple name.
   */
  public String getName();

  /**
   * Return the create agent responsible for derivative creation.
   */
  public CreateAgentI getCreateAgent();

  /**
   * Return the buy agent responsible for buying/selection.
   */
  public BuyAgentI getBuyAgent();

  /**
   * Return the deliver agent responsible for raw material delivery.
   */
  public DeliverAgentI getDeliveryAgent();

  /**
   * Return the finish agent responsible for raw material finishing.
   */
  public FinishAgentI getFinishAgent();}