```java
/* *********************************
 *   CreateAgent.java
 *     Handles the creation of new derivatives
 * *********************************/
package player.playeragent;

import player.*;
import edu.neu.ccs.demeterf.demfgen.lib.List;
import gen.*;

/** Class for creating a derivative */
public class CreateAgent implements PlayerI.CreateAgentI{

    /** Returns a newly created derivative of a different type than already existing derivatives */
    public Derivative createDerivative(Player player, List<Type> existing) {
        Type type = getFreshDerivType(existing);

        Derivative forSale = new Derivative(Util.freshName(player), player.id, new Price(1.0), type);

        forSale.price.val = IAmRobot.getBreakEven(forSale);
        return forSale;
    }

    private Type getFreshDerivType(List<Type> existing) {
        Type type;
        do {
                // Ensure this does not produce a relation >= 128, == 0, or odd (r mod 2 == 1)
                int relationNum = (Util.random(63) + 1) * 2;
                type = new Type(List.create(new TypeInstance(new RelationNr(relationNum))));
        } while (existing.contains(type));

        return type;
    }
}
```