```
/* *********************************
 *   CreateAgent.java
 *     Handles the creation of new derivitives
 * *********************************/
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 = Util.freshType(existing);
        Relation relation = new Relation(3, type.instances.top().r.v);
        double breakEven = Utils.getBreakEven(relation);

        //make sure the price stays in bounds
        if((breakEven + 0.001) <= 1)
        {
                breakEven += 0.001;
        }

        breakEven = Math.max(0.0, breakEven);
        breakEven = Math.min(1.0, breakEven);

        Price price = new Price(breakEven);
        return new Derivative(Util.freshName(player), player.id, price, type);
}*/
        public Derivative createDerivative(Player player, List<Type> existing){
        Type type = Util.freshType(existing);
        Derivative temp = new Derivative(null, player.id, null, type);
        double breakEven = Utils.getBreakEven(temp);

        //make sure the price stays in bounds
        if((breakEven + 0.001) <= 1)
        {
                breakEven += 0.001;
        }

        breakEven = Math.max(0.0, breakEven);
        breakEven = Math.min(1.0, breakEven);

        Price price = new Price(breakEven);
        return new Derivative(Util.freshName(player), player.id, price, type);
}
}
```