```
/* *********************************
 *   DeliverAgent.java
 *     Delivers Raw Materials
 * *********************************/
package player.playeragent;

import java.util.Date;
import java.util.Iterator;

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

/** Class for delivering raw material for a derivative */
public class DeliverAgent implements PlayerI.DeliverAgentI{
   static int vars = 10;

   /** Adds raw materials to the given derivatives */
   public Derivative deliverRawMaterial(Derivative needRM){
      RawMaterialInstance rmi = rawMaterialInst(needRM);
      Derivative gotRM = needRM.deliver(rmi);
      FinishAgent fagent = new FinishAgent();
      if(needRM.isClassic()) {
         return gotRM;
      }
      else
         return needRM.deliver(rmi.addSecret(fagent.bestRandomAssignment(gotRM)));
   }

   /** Compute a RawMaterial Instance for the given Derivative */
   private RawMaterialInstance rawMaterialInst(Derivative d){
      return new RawMaterialInstance(getConstraints(d));
   }

   /** Provides the array of constraints */
   private List<Constraint> getConstraints(Derivative d){
      Date d_start = new Date();
      long starttime = d_start.getTime();
      List<TypeInstance> list = d.type.instances;
      Iterator<TypeInstance> iter = list.iterator();
      List<Constraint> loc = List.<Constraint> create();
      Weight w;
      Variable[] v = new Variable[]{};
      RelationNr r;

      while(iter.hasNext()){
         r = iter.next().r;
         if(!Util.isOdd(r.v) && r.v < 127){
            w = new Weight(100);
            for (int x = 0; x <=vars; x++){
               for (int y = 0; y <= vars; y++){
                  if(x != y){
                     for(int z = 0; z <= vars; z++){
                        if(x != z && y!= z){
                           v = new Variable[] {(new Variable(new ident(("x" + x)))), (new Variable(new
ident(("x" + y)))), new Variable(new ident(("x" + z)))};
                           loc = loc.push(new Constraint(w, r, List.<Variable>create(v)));
                        }
                     }
                  }
               }
            }
         }

         else{
            w = new Weight(1);
            v = new Variable[] {(new Variable(new ident(("x" + 6)))), (new Variable(new ident(("x" + 7)
))), new Variable(new ident(("x" + 8)))};
            loc = loc.push(new Constraint(w, r, List.<Variable> create(v)));
         }

      }
      Date d_end = new Date();
      long endtime = d_end.getTime();
      System.out.println("Deliver Time: " + (endtime - starttime) + " ms");

      return loc;

   }
}
```