```java
package player;

import edu.neu.ccs.satsolver.MaxBiasOutput;
import edu.neu.ccs.satsolver.Polynomial;
import edu.neu.ccs.satsolver.SATSolverUtil;
import gen.RawMaterial;
import gen.Type;

/**
 * This class handles the bias and corresponding polynomials.
 */
public class Bias {
    private RawMaterial rm = null;
    private Type type = null;

    /**
     * Accept a raw material. Pay attention to weight. This is used by
     * FinishAgent.
     *
     * @param rm
     *            The raw material.
     */
    public Bias(RawMaterial rm) {
        this.rm = rm;
    }

    /**
     * Accept a type. Let all weights be 1.
     *
     * @param type
     *            The type.
     */
    public Bias(Type type) {
        this.type = type;
    }

    /**
     * Compute the appropriate maximum bias and the corresponding coefficients
     * of the polynomial for the CSP formula.
     *
     * @return A MaxBiasOutput object containing the maximum bias and
     *         polynomial.
     */
    public MaxBiasOutput max() {
        MaxBiasOutput result;
        InputInitial input;

        if (rm != null) {
            input = new InputInitial(rm);

            result = (MaxBiasOutput) SATSolverUtil.calculateBias(input);
        } else {
            input = new InputInitial(type);

            result = (MaxBiasOutput) SATSolverUtil.calculateBias(input);
        }

        return result;
    }

    /**
     * Use the maximum bias to calculate the break even price.
     *
     * @return The break even price.
     */
    public double breakEven() {
        MaxBiasOutput max = max();

        Polynomial poly = (Polynomial) max.getPolynomial();
        double bias = max.getMaxBias();

        double comp = poly.compute(bias);

        return comp;
    }
}
```