```java
package engine.pieces;

import chess.PieceType;
import chess.PlayerColor;
import engine.movements.*;
import engine.utils.Coordinates;

public abstract class Piece {
    private final PieceType type;
    private final PlayerColor color;
    private Movements movements;
    protected boolean firstMovement = true;

    public Piece(PieceType type, PlayerColor color, Movements movements) {
        if (type == null)
            throw new IllegalArgumentException("Piece type cannot be null");
        if (color == null)
            throw new IllegalArgumentException("Player color cannot be null");
        if (movements == null)
            throw new IllegalArgumentException("Movements cannot be null");

        this.type = type;
        this.color = color;
        this.movements = movements;
    }

    protected void setMovements(Movements movements) {
        this.movements = movements;
    }


    public PieceType getType() {
        return this.type;

    }

    public PlayerColor getColor() {
        return color;
    }

    public Coordinates[] getPossibleMovement(Coordinates positionInitial,
                                             Coordinates positionFinal) {
        Step possibleStep = this.movements.getPossibleStep(positionInitial,
                positionFinal);
        if (possibleStep == null) return null;

        Coordinates[] possibleMovement = possibleStep.getMovement();
        if (possibleMovement == null) return null;

        Coordinates[] m = new Coordinates[possibleMovement.length];
        for (int i = 0; i < possibleMovement.length; ++i)
            m[i] = new Coordinates(Coordinates.addition(possibleMovement[i],
                    positionInitial));

        return m;
    }

    public boolean movementIsOk(Coordinates positionInitial,
                                Coordinates positionFinal) {
        Coordinates[] possibleMovement = getPossibleMovement(positionInitial,
                positionFinal);
        if (possibleMovement == null) return false;

        for (Coordinates c : possibleMovement) {
            if (Coordinates.equal(c, positionFinal)) return true;
        }
        return false;
    }

    public boolean isFirstMovement() {
        return this.firstMovement;
    }

    public void clearFirstMovement() {
        this.firstMovement = false;
    }
}
```