```
package hw3;

import static api.Direction.*;

import api.Direction;
import api.Orientation;

/**
 * Represents a block in the Block Slider game.
 *  @author jcluse
 */
public class Block {

	private int firstRow;

	private int firstCol;

	private int length;

	private Orientation orientation;

	private int originalRow;

	private int originalCol;


	/**
	 * Constructs a new Block with a specific location relative to the board. The
	 * upper/left most corner of the block is given as firstRow and firstCol. All
	 * blocks are only one cell wide. The length of the block is specified in
cells.
	 * The block can either be horizontal or vertical on the board as specified
by
	 * orientation.
	 *
	 * @param firstRow    the first row that contains the block
	 * @param firstCol    the first column that contains the block
	 * @param length      block length in cells
	 * @param orientation either HORIZONTAL or VERTICAL
	 */
	public Block(int firstRow, int firstCol, int length, Orientation orientation)
{

		this.firstRow = firstRow;
		this.firstCol = firstCol;
		this.length = length;
		this.orientation = orientation;

		originalRow = firstRow;
		originalCol = firstCol;
	}

	/**
	 * Resets the position of the block to the original firstRow and firstCol
values
	 * that were passed to the constructor during initialization of the the
block.
	 */
	public void reset() {
```

```
            firstRow = originalRow;
            firstCol = originalCol;
      }

      /**
       * Move the blocks position by one cell in the direction specified. The
blocks
       * first column and row should be updated. The method will only move VERTICAL
       * blocks UP or DOWN and HORIZONTAL blocks RIGHT or LEFT. Invalid movements
are
       * ignored.
       *
       * @param dir direction to move (UP, DOWN, RIGHT, or LEFT)
       */
      public void move(Direction dir) {
            if (dir == UP && orientation == Orientation.VERTICAL) {
                  firstRow--;
            }
            else if (dir == DOWN && orientation == Orientation.VERTICAL) {
                  firstRow++;
            }
            else if (dir == RIGHT && orientation == Orientation.HORIZONTAL) {
                  firstCol++;
            }
            else if (dir == LEFT && orientation == Orientation.HORIZONTAL) {
                  firstCol--;
            }
      }

      /**
       * Gets the first row of the block on the board.
       *
       * @return first row
       */
      public int getFirstRow() {
            return firstRow;
      }

      /**
       * Sets the first row of the block on the board.
       *
       * @param firstRow first row
       */
      public void setFirstRow(int firstRow) {
            this.firstRow = firstRow;
      }

      /**
       * Gets the first column of the block on the board.
       *
       * @return first column
       */
      public int getFirstCol() {
            return firstCol;
      }

      /**
       * Sets the first column of the block on the board.
       *
```

```
         * @param firstCol first column
         */
        public void setFirstCol(int firstCol) {
              this.firstCol = firstCol;
        }

        /**
         * Gets the length of the block.
         *
         * @return length measured in cells
         */
        public int getLength() {
              return length;
        }

        /**
         * Gets the orientation of the block.
         *
         * @return either VERTICAL or HORIZONTAL
         */
        public Orientation getOrientation() {
              return orientation;
        }

        @Override
        public String toString() {
              return "(row=" + getFirstRow() + ", col=" + getFirstCol() + ", len=" +
getLength()
                          + ", ori=" + getOrientation() + ")";
        }
}
```