package com.twitter.servo.database

object Bitfield {

def multiValue(bits: Boolean\*): Int = {

bits.foldLeft(0) { (accum, bit) =>

(accum << 1) | (if (bit) 1 else 0)

}

}

def multiValueLong(bits: Boolean\*): Long = {

bits.foldLeft(0L) { (accum, bit) =>

(accum << 1) | (if (bit) 1L else 0L)

}

}

}

/\*\*

\* A mixin for unpacking bitfields.

\*/

trait Bitfield {

val bitfield: Int

/\*\*

\* Tests that a given position is set to 1.

\*/

def isSet(position: Int): Boolean = {

(bitfield & (1 << position)) != 0

}

/\*\*

\* takes a sequence of booleans, from most to least significant

\* and converts them to an integer.

\*

\* example: multiValue(true, false, true) yields 0b101 = 5

\*/

def multiValue(bits: Boolean\*): Int = Bitfield.multiValue(bits: \_\*)

}

trait LongBitfield {

val bitfield: Long

/\*\*

\* Tests that a given position is set to 1.

\*/

def isSet(position: Int): Boolean = {

(bitfield & (1L << position)) != 0

}

/\*\*

\* takes a sequence of booleans, from most to least significant

\* and converts them to a long.

\*

\* example: multiValue(true, false, true) yields 0b101 = 5L

\*/

def multiValue(bits: Boolean\*): Long = Bitfield.multiValueLong(bits: \_\*)

}