Bit

Nibble

Octet (Byte)

1/2/4/8/16-byte signed and unsigned integer

Character

public enum Boolean {False, True};

public enum Sign {Negative, Zero, Positive};

public interface Integer<Integer min, Integer max> {

public readonly Boolean Signed;

public readonly Sign Sign;

public readonly Int ByteCount;

public readonly Int BitCount {

return ByteCount \* 8;

}

}

- Sign {Negative, Zero, Positive}

- Signed / Unsigned

- WordSize / ByteCount

// or should these be methods of Array<Type<Bit>> ..

interface PowerOf2WordAligned extends Integer {

this ShiftLeft(Integer bits);

this ShiftRight(Integer bits);

this RotateLeft(Integer bits);

this RotateRight(Integer bits);

}

class Bit implements Integer<0, 1> {

}

class Nibble implements Integer<0, 15> {

readonly Char Hex {

if (this > 9) {

return ‘a’ + this – 10;

} else {

return ‘0’ + this;

}

}

}

class Byte implements Integer<0, 255>;

class TinyInt implements Integer<-32768, 32767>;

class Short implements Integer<-2147483648, 2147483647>;

Radix

StorageBits

PrecisionBits

MaximumExponent

Signed

+/-Infinity

+/-NaN

Byte

Short

Int

Long

Float

Double

Decimal