

The Zoo of Java Field Types (*Valhalla draft design, adding Extended Primitives*)

PRIMITIVES (one quadrant)
 PRIM = ¬ REF, scalar/extended primitive

REFERENCES (three quadrants)
 REF = ¬ PRIM, all are nullable, recursive

field-wise flattenable (like NOID)
 freely copyable (like NOID)
 no identity (like NOID)

defaults to all-zero bits
 no nulls, no direct recursion
 no construction safety
 no atomicity (tearing)
 no polymorphism

extended primitives, with class-like declaration
POSSIBLE IN VALHALLA

int long
 float double
 boolean char
 byte short

Island of Classless Scalar Primitives

their boxes

Memory Segment
 (value object: secure, no box, better VBC)

Optional
 (VBC migrated to value object, a NOID)

LocalDate
 (VBC = value based class, just before migration)

VALUE OBJECT REF (NOIDs)
 (includes PRIM boxes)
 freely copyable
 field-wise flattenable
 construction safety
 atomicity, no identity

THIS MIDDLE SWATH IS WHAT'S NEW IN VALHALLA

int[]
 (mutable array)

IDENTITY OBJECT REF (ID-O-SAURS)
 object identity
 construction safety
 fields stay together
 fast pointer compare

String
 (identity significant)

NON-CONCRETE REF
 polymorphic abstract class or interface

Cursor

Value Object (marker)

List

Identity Object (marker)

Object

Iterator

Abstract List
 (partially concrete)

HashMap
 (mutable object)

List
 Cursor
 its box

Point
 its box

The Zoo of Java Field Types (*Valhalla draft design, Value Objects only*)

PRIMITIVES (one quadrant)

PRIM = \neg REF, scalar primitives only

REFERENCES (three quadrants)

REF = \neg PRIM, all are nullable, recursive

field-wise flattenable
freely copyable
no identity
defaults to all-zero bits
no nulls
uncertain 64-bit atomicity
no polymorphism

`int long`

`float double`

`boolean char`

`byte short`

Integer
Long, etc.
(primitive boxes)

VALUE OBJECT
REF (NOIDs)
freely copyable
field-wise flattenable
construction safety
atomicity, no identity

Optional
(VBC migrated
to value object,
a NOID)

Memory
Segment
(value object:
secure, no box,
better VBC)

LocalDate
(VBC = value
based class,
just before
migration)

`int[]`
(mutable
array)

IDENTITY
OBJECT REF
(ID-O-SAURs)
object identity
construction safety
fields stay together
fast pointer compare

String
(identity
significant)

NON-CONCRETE REF
polymorphic abstract class or interface

Cursor

Value
Object
(marker)

List

List
Cursor

Point

Object

Identity
Object
(marker)

Iterator

Abstract
List
(partially
concrete)

HashMap
(mutable
object)

The Zoo of Java Field Types (*before Valhalla*)

PRIMITIVES (one quadrant)

PRIM = \neg REF, scalar primitives only

REFERENCES (three quadrants)

REF = \neg PRIM, all are nullable, recursive

field-wise flattenable
freely copyable
no identity

defaults to all-zero bits
no nulls
uncertain 64-bit atomicity
no polymorphism

`int long`

`float double`

`boolean char`

`byte short`

`Integer
Long, etc.`

(primitive boxes)

`LocalDate`

(VBC = value based class)

`Optional`

(VBC = value based class)

`Memory Segment`

(VBC = value based class)

VBCs = value based classes
non-VBCs

NON-CONCRETE REF

polymorphic abstract class or interface

`List`

`Object`

`Iterator`

CONCRETE OBJECT REFS

object identity
construction safety
fields stay together
fast pointer compare

`Abstract List`

(partially concrete)

`HashMap`

(mutable object)

`int[]`

(mutable array)

`String`

(identity significant)