

Just start (with Value Objects)

Katharina Damschen

Software Architect and Consultant

factor 10



A top-down view of a cluttered wooden desk. In the center is a silver laptop with a blue screen. To its left is another silver laptop, and to its right is a third silver laptop with a blue screen. A person's hands are visible at the top right, typing on the third laptop. A person's hands are visible at the bottom right, holding a smartphone. Various other items are scattered on the desk: a black teapot, a glass of water, a bowl of snacks, a pair of headphones, a small orange device, a black bag, and several papers. The desk is made of light-colored wood.



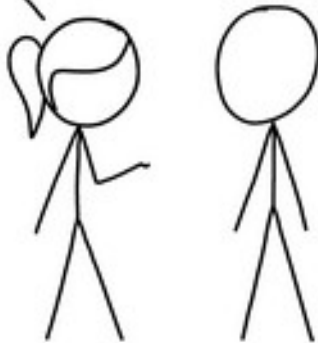
?





SILICATE CHEMISTRY IS SECOND NATURE TO US GEOCHEMISTS, SO IT'S EASY TO FORGET THAT THE AVERAGE PERSON PROBABLY ONLY KNOWS THE FORMULAS FOR OLIVINE AND ONE OR TWO FELDSPARS.

AND QUARTZ, OF COURSE.
OF COURSE.



EVEN WHEN THEY'RE TRYING TO COMPENSATE FOR IT, EXPERTS IN ANYTHING WILDLY OVERESTIMATE THE AVERAGE PERSON'S FAMILIARITY WITH THEIR FIELD.





Knitting Project

Yarn



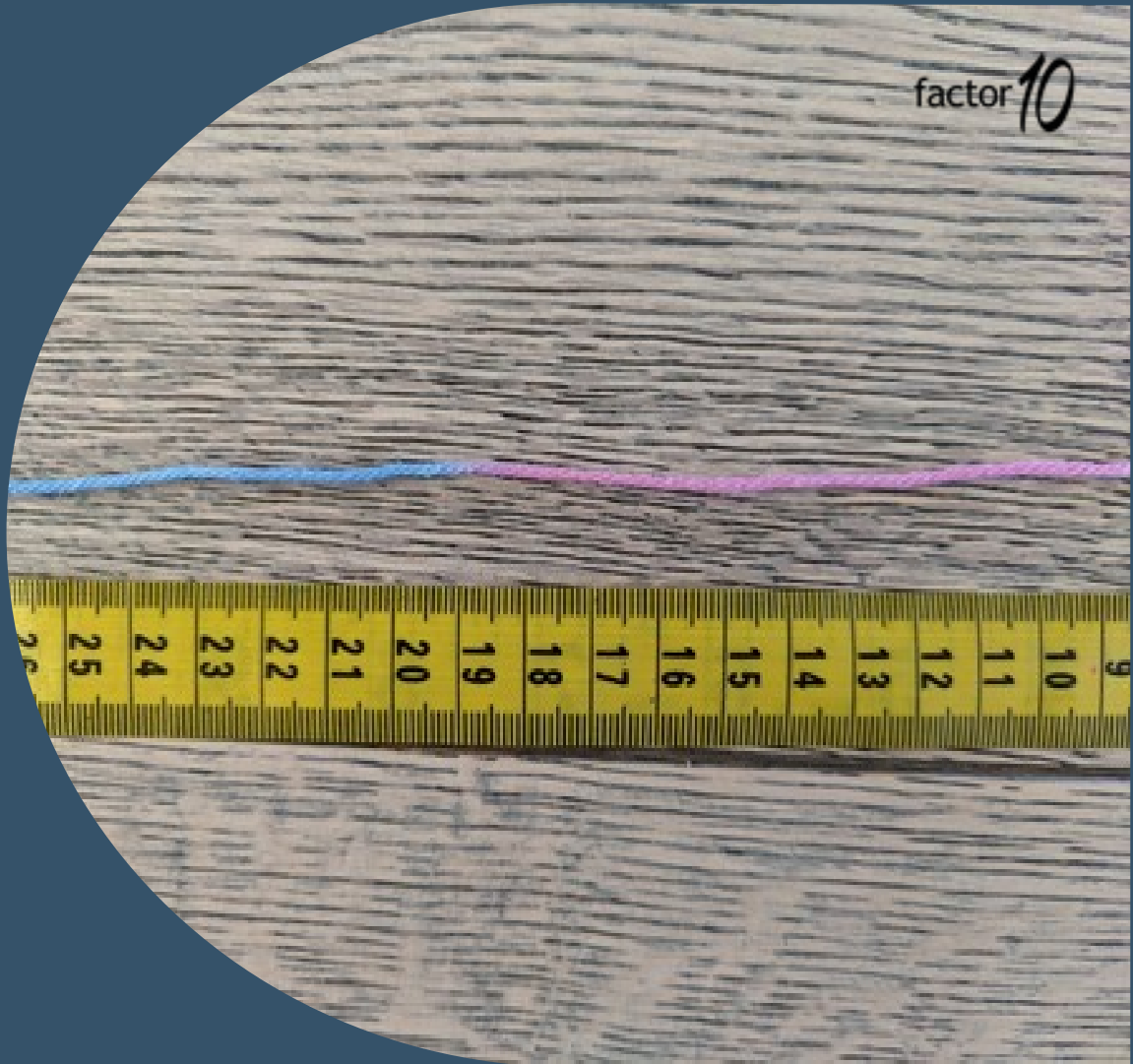
factor 10



Skein

factor 10

Yardage



The task

factor 10

Knitting Proj.

<https://www.knitting-project.ru/knity-project>

My knitting project



Pattern

Some Pattern

Knitting Needle Size

4.0 mm

Yarn Name

Some Yarn

Yardage per Skein

200

Number of skeins

5

Available Yardage

??

Suggested yarn

To knit this top, you'll need fingering weight yarn (I recommend a yarn that doesn't stretch too much).

For size A (B, C, D, E) F (G, H, I, J) you'll need 490 (530, 600, 670, 730) 800 (870, 950, 1120, 1200) meters of yarn.

Please, have in mind that the yarn needed may vary depending on how long you want your top.

The 'solution' ?

factor 10

```
object YardageCalculator {  
    fun calculateTotalYardage(yardagePerSkein: Long, skeinCount: Int) =  
        yardagePerSkein * skeinCount  
}
```

@Test

```
fun `calculate total`() {  
    val total = YardageCalculator.calculateTotalYardage(200, 5)  
  
    assertEquals(1000, total)  
}
```



Event Storming?

Ubiquitous Language?

Domain-driven Design

Domain Model?

Context Map?

Tactical Patterns?

Domain-driven Design

Building Blocks
And Patterns



Entities

factor 10

Objects that
are defined
primarily by
their identity
(not by their
attributes).

```
class KnittingProject(val id: UUID = UUID.randomUUID(),  
                      var yardagePerSkein: Long,  
                      var skeinCount: Int  
) {  
    override fun equals(other: Any?): Boolean {  
        if (this === other) return true  
        if (other !is KnittingProject) return false  
        return id == other.id  
    }  
    override fun hashCode(): Int = id.hashCode()  
}
```

```
// val total = project.yardagePerSkein * project.skeinCount
```

Anemic Models

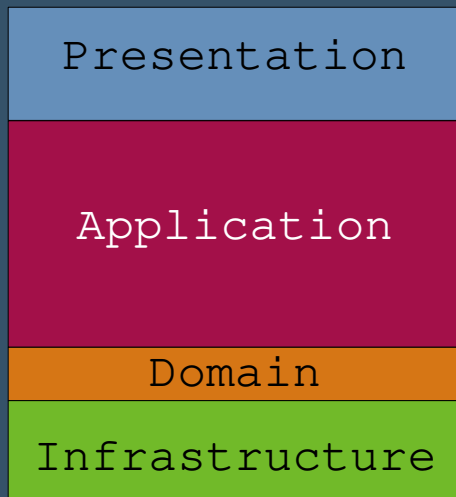
factor 10

- Domain objects are just data containers.
- No encapsulation or invariant protection.
- Services contain all business rules.
- Overreliance on Entities.

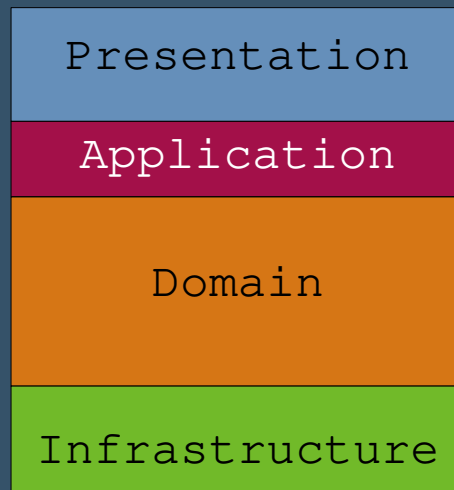
The Domain Layer

factor 10

Layered Architecture



Anemic Domain Models
Transaction Script



Domain Driven



Value Objects^{factor 10}

Objects without
a conceptual
identity that
are primarily
defined by their
attributes.

Immutable objects



objects whose
state cannot be
changed after
their creation.

Value Objects

factor 10

```
data class Skein(val yardage: Long) {  
    init {  
        require(yardage >= 0) { "Yardage must be positive" }  
    }  
}
```

No
general
rules



Bounded Context

factor 10

Context in which
a domain model is
valid.

Ubiquitous Language^{factor 10}

Yoke

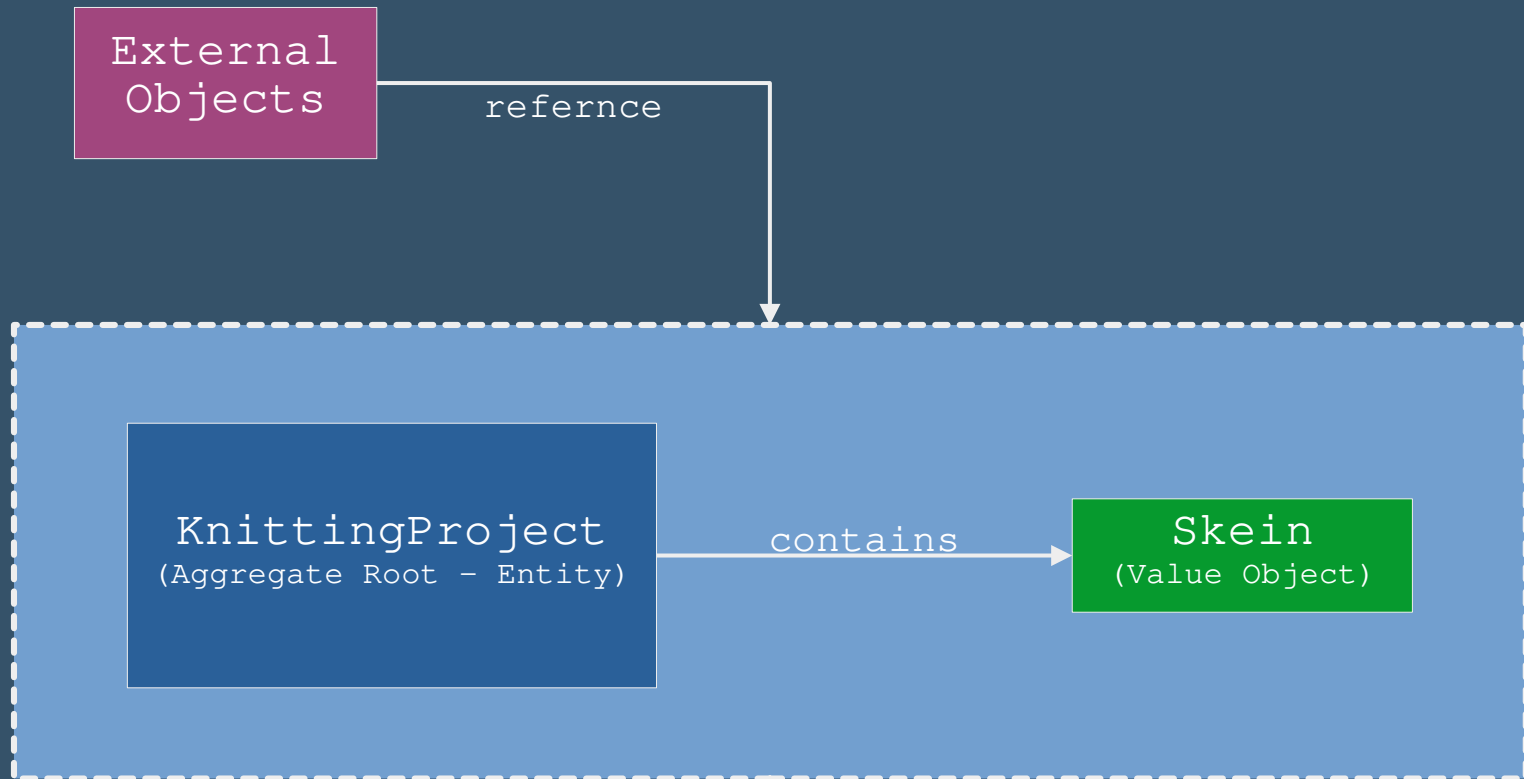
CO	36 (42, 56, 62, 68) sts with Color A
Row 1 (WS)	p1 (1, 3, 4, 5), pm, p4 (6, 8, 8, 8), pm, p26 (28, 34, 38, 42), pm, p4 (6, 8, 8, 8), pm, p1 (1, 3, 4, 5)
Row 2 (RS)	*k to m, m1r, sm, k1, m1l, k to 1 st before m, m1r, k1, sm, m1l* repeat twice in total, k to end (+ 8 sts)
Row 3	purl all stitches, slip markers as you pass them switch color
Row 4	k1, m1l,*k to m, m1r, sm, k1, m1l, k to 1 st before m, m1r, k1, sm, m1l* repeat twice in total, k to last st, m1r, k1 (+ 10 sts)
Row 5	purl all stitches, slip markers as you pass them
Repeat	<i>Rows 2 - 5</i> 11 (12, 13, 14, 15) more times, changing the yarn color after every 4th row (=37 (40, 45, 49, 53) front sts each front, 52 (58, 64, 68, 72) sleeve sts each sleeve and 74 (80, 90, 98, 106) back sts, in total 252 (276, 308, 332, 356) sts)



Aggregate^{factor 10}

Cluster of
associated
objects,
treated as a
unit for the
purpose of data
changes

Aggregate



Putting it all
together

```
class KnittingProject private constructor(val id: UUID,  
                                          val skeins: List<Skein>) {  
    constructor(id: UUID = UUID.randomUUID()) : this(id, emptyList())  
  
    fun addSkeins(skein: Skein, amount: Int): KnittingProject {  
        require(amount > 0) { "Amount must be positive" }  
        val newSkeins = List(amount) { skein }  
        return KnittingProject(id, skeins + newSkeins)  
    }  
  
    fun calculateTotalAvailableYardage(): Long =  
        skeins.sumOf { it.yardage }  
} // equals & hashCode unchanged
```

Value Objects
everywhere!

No simple
types in the
domain model.



```
data class Skein(val yardage: Long) {  
    init {  
        require(yardage >= 0) { "Yardage must be positive" }  
    }  
}
```

```
data class Skein(val yardage: Yardage)
```

```
value class Yardage(val value: Long) {  
    init {  
        require(value >= 0) { "Yardage must be positive." }  
    }  
}
```

```
value class Yardage(val value: Long) {  
    init {  
        val result = validate(value)  
        require(result is ValidationResult.Success) { (result as ValidationResult.Failure).error }  
    }  
  
    companion object {  
        fun validate(value: Long): ValidationResult {  
            return when {  
                value < 0 ->  
                    ValidationResult.Failure("Yardage must be positive.")  
  
                else -> ValidationResult.Success  
            }  
        }  
    }  
}
```

// use a result wrapper to be able to return
// a specific message, not just true or false
sealed interface ValidationResult {
 object Success : ValidationResult
 data class Failure(val error: String) : ValidationResult
}

```
companion object {  
    private const val MIN_YARDAGE = 0L  
    private const val MAX_YARDAGE = 50000L  
  
    fun validate(value: Long): ValidationResult {  
        return when {  
            value < MIN_YARDAGE ->  
                ValidationResult.Failure("Yardage must be positive.")  
  
            value > MAX_YARDAGE ->  
                ValidationResult.Failure("Yardage must not exceed $MAX_YARDAGE.")  
  
            else -> ValidationResult.Success  
        }  
    }  
}
```

Screenshot of ravelry.com

Weight	DK (11 wpi) ?
Wraps per inch	
Yardage	273 yards (250 meters)
Unit weight	100 grams (3.53 ounces)
Gauge	21.0 to 24.0 sts = 4 inches
Needle size	US 5 - 7 or 3.75 - 4.5mm
Hook size	4.5mm - 5.5mm (I)
Fibers	100% Wool - Rambouillet
Texture	plied
Attributes	Care: Dry Flat, Hand Wash Cold Color: Solid color Dye: Machine dyed Put up: Winding required
Yarn origin	Milled: Italy

@Test

```
fun `calculate total`() {  
    val total = YardageCalculator.calculateTotalYardage(200, 5)  
  
    assertEquals(1000, total)  
}
```

@Test

```
fun `calculateTotalAvailableYardage for five skeins returns correct yardage`() {  
    val skein1 = Skein(Yardage(200, LengthUnit.Meters))  
    val sweater = KnittingProject().addSkeins(skein1, 5)  
  
    val actualYardage = sweater.calculateTotalAvailableYardage()  
  
    assertEquals(Yardage(1000, LengthUnit.Meters), actualYardage)  
}
```

Lessons from real projects



```
public class Project  
{  
    public string Name { get; }
```

```
...
```

198910062382

info@example.com

070-1740699



Advantages

- Immutability.
- Type safety.
- Encapsulation of validation and business logic.
- Better understandability and testability.
- Explicit domain concepts in code.
- A possibility to start asking questions.
- Simple classes.
- Accessible entry point into DDD.

Getting Started

- Identify potential Value Objects.
- Validate them.
- Make them immutable.
- Implement equality.



Thank you!

`katharina.damschen.net`

`katharina.damschen@factor10.com`

`linkedIn: katharina-damschen`

`mastodon.nu/@katharina`