

Ikarus::KirchhoffLoveShell
::calculateVectorImpl

Ikarus::LinearElastic
::calculateVectorImpl

Ikarus::NonLinearElastic
::calculateVectorImpl

Ikarus::Volume::calculate
VectorImpl

```
graph LR; A[Ikarus::KirchhoffLoveShell::calculateVectorImpl] --> D[Ikarus::Volume::calculateVectorImpl]; B[Ikarus::LinearElastic::calculateVectorImpl] --> D; C[Ikarus::NonLinearElastic::calculateVectorImpl] --> D;
```

The diagram illustrates a design pattern where three different classes (KirchhoffLoveShell, LinearElastic, and NonLinearElastic) implement a common interface or base class (Volume). Each class has its own implementation of the calculateVectorImpl method, which is represented by a box. Arrows point from each of these three boxes to a single box on the right, which represents the Volume class's calculateVectorImpl method. This suggests that the Volume class is the base or interface, and the other three classes are concrete implementations that inherit from or implement the Volume class's method.