```julia
"""
    struct GeneralSpace{𝕜} <: ElementarySpace{𝕜}

A finite-dimensional space over an arbitrary field `𝕜` without additional
structure. It is
thus characterized by its dimension, and whether or not it is the dual and/or
conjugate
space. For a real field `𝕜`, the space and its conjugate are the same.
"""
struct GeneralSpace{𝕜} <: ElementarySpace{𝕜}
    d::Int
    dual::Bool
    conj::Bool
    function GeneralSpace{𝕜}(d::Int, dual::Bool, conj::Bool) where {𝕜}
        d >= 0 ||
            throw(ArgumentError("Dimension of a vector space should be bigger than
zero"))
        if 𝕜 isa Field
            new{𝕜}(Int(d), dual, (𝕜 ⊆ ℝ) ? false : conj)
        else
            throw(ArgumentError("Unrecognised scalar field: $𝕜"))
        end
    end
end
GeneralSpace{𝕜}(d::Int = 0; dual::Bool = false, conj::Bool = false) where {𝕜} =
    GeneralSpace{𝕜}(d, dual, conj)

dim(V::GeneralSpace) = V.d
isdual(V::GeneralSpace) = V.dual
isconj(V::GeneralSpace) = V.conj

Base.axes(V::GeneralSpace) = Base.OneTo(dim(V))

dual(V::GeneralSpace{𝕜}) where {𝕜} =
    GeneralSpace{𝕜}(dim(V), !isdual(V), isconj(V))
Base.conj(V::GeneralSpace{𝕜}) where {𝕜} =
    GeneralSpace{𝕜}(dim(V), isdual(V), !isconj(V))

function Base.show(io::IO, V::GeneralSpace{𝕜}) where {𝕜}
    if isconj(V)
        print(io, "conj(")
    end
    print(io, "GeneralSpace{", 𝕜, "}(", dim(V), ")")
    if isdual(V)
        print(io, "'")
    end
    if isconj(V)
        print(io, ")")
    end
```

```
end
```