

Class `CodeList<E extends CodeList<E>>`

[Object](#)

`CodeList<E>`

Type Parameters:

E - The type of this code list.

All Implemented Interfaces:

[Serializable](#), [Comparable<E>](#)

Direct Known Subclasses:

[AssociationType](#), [AxisDirection](#), [BandDefinition](#), [CellGeometry](#), [CharacterSet](#), [Classification](#), [Context](#), [CoverageContentType](#), [Datatype](#), [DateType](#), [Dimension-NameType](#), [EvaluationMethodType](#), [GeometricObjectType](#), [GeometryType](#), [Imaging-Condition](#), [InitiativeType](#), [KeywordType](#), [MaintenanceFrequency](#), [MediumFormat](#), [MediumName](#), [ObjectiveType](#), [Obligation](#), [OnLineFunction](#), [OperationType](#), [PixelIn-Cell](#), [PixelOrientation](#), [PolarizationOrientation](#), [PositionableProjection](#), [PresentationForm](#), [Priority](#), [Progress](#), [RangeMeaning](#), [Restriction](#), [Role](#), [Scope-Code](#), [Sequence](#), [SpatialRepresentationType](#), [TopicCategory](#), [TopologyLevel](#), [TransferFunctionType](#), [Trigger](#), [VerticalDatumType](#)

```
@UML(identifier="CodeList",
      specification=ISO_19103)
public abstract class CodeList<E extends CodeList<E>>
extends Object
implements Comparable<E>, Serializable
```

Base class for all code lists. Subclasses shall provides a `values()` method which returns all `CodeList` element in an array of the appropriate class.

Code lists are extensible, i.e. invoking the `valueOf(String)` method in any subclass will automatically add the newly created `CodeList` element in the array to be returned by `values()`.

Since:

1.0

See Also:

[Serialized Form](#)

Nested Class Summary

Nested Classes

Modifier and Type	Class	Description
-------------------	-------	-------------

static interface **CodeList.Filter** Used by `valueOf(Class, Filter)` to select codes matching an arbitrary criterion.

Constructor Summary

Constructors

Modifier	Constructor	Description
protected	CodeList (String [↗] name, Collection [↗] < E > values)	Creates a new code list element and add it to the given collection.

Method Summary

All Methods

Static Methods

Instance Methods

Abstract Methods

Concrete Methods

Modifier and Type	Method	Description
final int	compareTo (E other)	Compares this code with the specified object for order.
abstract E []	family ()	Returns the list of codes of the same kind as this code.
String [↗]	identifier ()	Returns the identifier declared in the UML annotation, or null if none.
final String [↗]	name ()	Returns the programmatic name of this code list constant.
String [↗] []	names ()	Returns all the names of this code.
final int	ordinal ()	Returns the ordinal of this code constant.
protected Object [↗]	readResolve ()	Resolves the code list to an unique instance after deserialization.
String [↗]	toString ()	Returns a string representation of this code list.

<code>static <T extends Code-</code> <code>List<T>></code> <code>T</code>	<code>valueOf(Class<T> codeType,</code> <code>String name)</code>	Returns the code of the given type that matches the given name, or returns a new one if none match it.
---	--	--

<code>static <T extends Code-</code> <code>List<T>></code> <code>T</code>	<code>valueOf(Class<T> codeType,</code> <code>CodeList.Filter filter)</code>	Returns the code of the given type that matches the given criterion, or returns a new one if none match it.
---	---	---

Methods inherited from class `Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

Constructor Details

`CodeList`

```
protected CodeList(String name,  
                    Collection<E> values)
```

Creates a new code list element and add it to the given collection. Subclasses will typically give a static reference to an `ArrayList` for the values argument. This list is used for `values()` method implementations.

Parameters:

name - The code name.

values - The collection to add the element to.

Method Details

`valueOf`

```
public static <T extends CodeList<T>> T valueOf(Class<T> codeType,  
                                                String name)
```

Returns the code of the given type that matches the given name, or returns a new one if none match it. More specifically, this methods returns the first element of the given class where `name().equals(name)` returned true. If no such element is found, then a new instance is created using the constructor expecting a single `String` argument.

Implementation note: The `codeType` class needs to be initialized before to invoke this method. This is usually the case when the caller is a static method of the `codeType` class.

However in other situations, callers may need to initialize explicitly the given class.

Type Parameters:

T - The compile-time type given as the `codeType` parameter.

Parameters:

`codeType` - The type of code list.

`name` - The name of the code to obtain, or `null`.

Returns:

A code matching the given name, or `null` if the name is `null`.

valueOf

```
public static <T extends CodeList<T>> T valueOf(Class<T> codeType,  
                                              CodeList.Filter filter)
```

Returns the code of the given type that matches the given criterion, or returns a new one if none match it. More specifically, this methods returns the first element (in declaration order) of the given class where `filter.accept(code)` returns `true`. If no such element is found, then there is a choice:

- If `CodeList.Filter.codename\(\)` returns `null`, then this method returns `null`.
- Otherwise a new instance is created using the constructor expecting a single [String](#) argument, which is given the value returned by `codename()`.

Type Parameters:

T - The compile-time type given as the `codeType` parameter.

Parameters:

`codeType` - The type of code list.

`filter` - The criterion for the code to obtain.

Returns:

A code matching the given criterion, or `null` if there is no match and `CodeList.Filter.codename\(\)` returns `null`.

Since:

2.3

family

```
public abstract E[] family()
```

Returns the list of codes of the same kind as this code. This is similar to the static `values()` method provided in `CodeList` subclasses, except that `family()` does not require the class to be known at compile-time - provided that at least one instance of the family is available. The

`static values()` method has the opposite constraints (does not require a code instance, but the class needs to be known at compile time unless [reflection](#) is used).

Returns:

The codes of the same kind as this code.

names

```
public String[] names()
```

Returns all the names of this code. The returned array contains the following elements, with duplicated values and null values removed:

- The programmatic [name](#)
- The UML [identifier](#)
- The [charset](#) name (CharacterSet code list only)

Those names are typically equal except for the case (programmatic names are upper case while UML names are lower case) and special characters like ' - '.

Returns:

All names of this code constant. This array is never null and never empty.

Since:

2.3

name

```
public final String name()
```

Returns the programmatic name of this code list constant. This is the name of the public static field which declare the code.

Returns:

The name of this code constant.

identifier

```
public String identifier()
```

Returns the identifier declared in the [UML](#) annotation, or null if none. The UML identifier shall be the ISO or OGC name for this code constant.

Returns:

The ISO/OGC identifier for this code constant, or null if none.

Since:

2.2

ordinal

```
public final int ordinal()
```

Returns the ordinal of this code constant. This is its position in its elements declaration, where the initial constant is assigned an ordinal of zero.

Returns:

The position of this code constants in elements declaration.

compareTo

```
public final int compareTo(E other)
```

Compares this code with the specified object for order. Returns a negative integer, zero, or a positive integer as this object is less than, equal to, or greater than the specified object.

Code list constants are only comparable to other code list constants of the same type. The natural order implemented by this method is the order in which the constants are declared.

Specified by:

`compareTo` in interface `Comparable`[↗]`<E` extends `CodeList``<E>>`

Parameters:

`other` - The code constant to compare with this code.

Returns:

A negative value if the given code is less than this code, a positive value if greater or 0 if equal.

toString

```
public String↗ toString()
```

Returns a string representation of this code list.

Overrides:

`toString`[↗] in class `Object`[↗]

readResolve

```
protected Object↗ readResolve()  
                throws ObjectStreamException↗
```

Resolves the code list to an unique instance after deserialization. The instance is resolved using its `name` only (not its `ordinal`).

Returns:

This code list as an unique instance.

Throws:

[ObjectStreamException](#)[↗] - if the deserialization failed.