

NAME

oo::class -
class of all classes

SYNOPSIS

```
package require TclOO
```

```
oo::class method ?arg ...?
```

CLASS HIERARCHY

oo::object -> **oo::class**

DESCRIPTION

Classes are objects that can manufacture other objects according to a pattern stored in the factory object (the class). An instance of the class is created by calling one of the class's factory methods, typically either **create** if an explicit name is being given, or **new** if an arbitrary unique name is to be automatically selected.

The **oo::class** class is the class of all classes; every class is an instance of this class, which is consequently an instance of itself. This class is a subclass of **oo::object**, so every class is also an object. Additional metaclasses (i.e., classes of classes) can be defined if necessary by subclassing **oo::class**. Note that the **oo::class** object hides the **new** method on itself, so new classes should always be made using the **create** method.

CONSTRUCTOR

The constructor of the **oo::class** class takes an optional argument which, if present, is sent to the **oo::define** command (along with the name of the newly-created class) to allow the class to be conveniently configured at creation time.

DESTRUCTOR

The **oo::class** class does not define an explicit destructor. However, when a class is destroyed, all its subclasses and instances are also destroyed, along with all objects that it has been mixed into.

EXPORTED METHODS

cls **create** *name* ?*arg* ...?

This creates a new instance of the class *cls* called *name* (which is resolved within the calling context's namespace if not fully qualified), passing the arguments, *arg* ..., to the constructor, and (if that returns a successful result) returning the fully qualified name of the created object (the result of the constructor is ignored). If the constructor fails (i.e. returns a non-OK result) then the object is destroyed and the error message is the result of this method call.

cls **new** ?*arg* ...?

This creates a new instance of the class *cls* with a new unique name, passing the arguments, *arg* ..., to the constructor, and (if that returns a successful result) returning the fully qualified name of the created object (the result of the constructor is ignored). If the constructor fails (i.e., returns a non-OK result) then the object is destroyed and the error message is the result of this method call.

Note that this method is not exported by the **oo::class** object itself, so classes should not be created using this method.

NON-EXPORTED METHODS

The **oo::class** class supports the following non-exported methods:

cls **createWithNamespace** *name nsName* ?*arg* ...?

This creates a new instance of the class *cls* called *name* (which is resolved within the calling context's namespace if not fully qualified), passing the arguments, *arg* ..., to the constructor, and (if that returns a successful result) returning the fully qualified name of the created object (the result of the constructor is ignored). The name of the instance's internal namespace will be *nsName* unless that namespace already exists (when an arbitrary name will be chosen instead). If the constructor fails (i.e., returns a non-OK result) then the object is destroyed and the error message is the result of this method call.

EXAMPLES

This example defines a simple class hierarchy and creates a new instance of it. It then invokes a method of the object before destroying the hierarchy and showing that the destruction is transitive.

```
oo::class create fruit {  
    method eat {} {  
        puts "yummy!"  
    }  
}  
oo::class create banana {  
    superclass fruit  
    constructor {} {  
        my variable peeled  
        set peeled 0  
    }  
    method peel {} {  
        my variable peeled  
        set peeled 1  
        puts "skin now off"  
    }  
    method edible? {} {  
        my variable peeled  
        return $peeled  
    }  
    method eat {} {  
        if {[my edible?]} {  
            my peel  
        }  
        next  
    }  
}  
set b [banana new]  
$b eat           -> prints "skin now off" and "yummy!"  
fruit destroy  
$b eat           -> error "unknown command"
```

SEE ALSO

[oo::define\(n\)](#), [oo::object\(n\)](#)

KEYWORDS

[class](#), [metaclass](#), [object](#)