You need to declare generic functions in S4 before you can define methods for them. If no definition exists you will see the following error:

> setClass("track", slots = c(x="numeric", y = "numeric"))

> setMethod("foo", signature(x = "track"))

Error in setMethod("foo", signature(x = "track"), definition = function(x) cat(x)) :

no existing definition for function ‘foo’

Generic functions are declared with the setGeneric() function, which must precede the call to setMethod():

> setClass("track", slots = c(x="numeric", y = "numeric"))

> setGeneric("foo", def = str)

[1] "foo"

> setMethod("foo", signature(x = "track"), definition = function(x) cat(x))

But when you develop an R package you may have several classes that define their own methods for the same generic function. So where to put the definition of the generic function? One solution is to put all your classes in the same source file, and have the call to setGeneric() precede all calls to setMethod(). This will work but maintainability dictates that each class should belong to its own source file, named R/.R.

Instead, you might consider placing all calls to setGeneric() in a source file guaranteed to be loaded before all other files. For example, you might call that file \_\_generics.R, which will be loaded first because when R loads a package it reads the source files in alphabetical order. This will work too but there’s a more elegant way.

[R will read the source files in alphabetical order](https://cran.r-project.org/doc/manuals/r-release/R-exts.html#The-DESCRIPTION-file) unless a Collate field in the DESCRIPTION file says otherwise. That field lets you specify the order in which you want your source files to be loaded. If present, it must list **all** source files.

Maintaining such a field for more than about 5 source files can quickly become tedious. Fortunately, the roxygen2 package [has a little-known feature](https://cran.r-project.org/web/packages/roxygen2/vignettes/collate.html) which will generate the Collate field for you. If you use the @include tag anywhere in a file, roxygen will generate a Collate field which includes all source files, in such an order that all included files are listed first.

For example, if you have a file R/generics.R with the following declaration:

setGeneric("foo", ...)

and your MyClass needs to define that generic, use the following:

#' @include generics.R

MyClass <- setClass(...)

setMethod("foo", ...)

When you now run roxygen2 (and don’t forget this step, it won’t be done automatically for you) it will generate a Collate field with all your R code files, correctly sorted.