Institut für Nachrichtentechnik
und Hochfrequenztechnik
G. Doblinger 2.11.2009

How to structure a MATLAB project?

1. All functions in different m-files in the same directory
2. A main function in an m-file containing subfunctions
3. A main function in the parent directory, and other functions used in a subdirectory called `private`
4. Functions organized as above plus class definitions in class directories starting with character `@` like `@myfirstclass`
5. Use of packages with a package parent directory name starting with character `+` like `+mypackage`

Method 1

We create a directory and put all functions used by the project into this directory. Very simple but function names must not be coincide with other functions. If you do not know that e.g. function `sin` exists, then try `which sin` in the MATLAB command window. This method should be used for small projects.

Method 2

We create an m-File with the function name and put all functions (called subfunctions) in the m-File after the main function.

Example of an m-File named `myfunction.m`:

```
function y = myfunction(a,b,c)
% ...
y = mysubfunction1(a);   % call of a subfuntion
% ...
end   % end of myfunction (can be omitted)

function y1 = mysubfunction1(a)
% ...
end   % end of mysubfunction1 (can be omitted)

function y2 = mysubfunction2(a,b)
% ...
end   % end of mysubfunction2 (can be omitted)
```

Method 3

We create a directory containing the main function and other functions used by the project. Additional functions which are needed by all functions of the project are put into a subdirectory called `private`. Files in this private directory are known to project functions in the parent directory only.

Example directory structure:

```
myproject
    |_ mainfunction.m
    |_ function1.m
    |_ private
          |_ function2.m
          |_ function3.m
```

Functions `function2` and `function3` may be used in `mainfunction` and `function1` but not outside of directory `myproject`.

Method 4

We apply one of the above methods and additionally use classes of object-oriented programming as discussed earlier in this course.

Example directory structure:

```
myproject
    |_ mainfunction.m
    |_ function1.m
    |_ private
          |_ function2.m
          |_ function3.m
    |_ @class1
          |_ class1.m
    |_ @class2
          |_ class2.m
```

Method 5

MATLAB packages may be used with large projects where we want to encapsulate all files and classes to a self-contained application. Function and class names need to be unique within the package only. Functions and classes defined within a package can also be used outside the package. The (parent) package directory must be included in the MATLAB path. Package directory names are starting with character `+`.

Example directory structure:

```
+mypackage
    |_ mainfunction.m
    |_ function1.m
```

```
    |_ private
          |_ function2.m
          |_ function3.m
    |_ @class1
          |_ class1.m
    |_ @class2
          |_ class2.m
    |_ +subpackage1
          |_ ...
```

Functions used inside and/or outside the package directory must be referenced by the package name like `y = mypackage.mainfunction(x)`. The same holds with classes as in the example `obj = mypackage.class1(a,b,c)`.

As an alternative, we can import package functions and classes:

Example of package class import within a function:

```
function y = function1(x)
   import mypackage.class1

   obj = class1(a,b,c);  % call constructor of imported class
   ...
end
```

Example of package function import within a function:

```
function y = mainfunction(x,a,b,c)
   import mypackage.function1

   y = function1(x);     % call imported function
   ...
end
```