

Programming (ERIM)

Lecture 4: Side effects, functions and procedures

Tommi Tervonen

Econometric Institute, Erasmus School of Economics

Counting sum of the elements of an integer array

```
public static int countSum(int[] array) {  
    for (int i=1;i<array.length;i++) {  
        array[i] = array[i] + array[i-1];  
    }  
    return array[array.length-1];  
}
```

Counting sum of the elements of an integer array

```
public static int countSum(int[] array) {  
    for (int i=1;i<array.length;i++) {  
        array[i] = array[i] + array[i-1];  
    }  
    return array[array.length-1];  
}
```

- Returns the correct value, but also modifies the parameter array as a *side effect*.
- What would you expect from:

```
public static int countSum(int[] array)
```

- Unexpected side effects make code difficult to understand
- There are also *desired* side effects, e.g. sorting the contents of an array
- Other desired side effects: printing to console, showing a pop-up window, writing to a file

Parameter passing schemes

- For side effects to be possible, parameters have to be passed *by reference*: only a reference (memory address) of the variable is passed to the called method
- Other main technique for parameter passing is to pass *by value*: a local copy of the variable is created within the called method

```
f <- function(a) {  
  a <- a + 2  
  a  
}  
## Pass by value  
x <- 5  
y <- f(x) # x == 5  
## Pass by reference  
x <- 5  
y <- f(x) # x == 7
```

- In imperative programming we classify methods into
 - **Functions**, that return a value but do not alter the parameters in any way
 - **Procedures**, that alter some of the parameters but do not return a value

```
sort(c(2, 3, 4))  
message("hello")  
asum <- sum(c(2, 3, 4))
```

Parameter passing in Matlab and R

- Matlab and R pass everything by value
- Matlab: Matrices are passed by reference until they are modified the first time, at which point a local copy is created (!)
- The Matlab OO-extension allows to pass references by value by using the handle class (but don't use it)

There are no procedures: only functions

There are no procedures: only functions

- Pros:
 - No undesired side effects
- Cons:
 - No desired side effects
 - Many algorithms can be expressed more clearly with procedures
 - Recursive algorithms become slow without procedures

```
function A = sort(A)
    leftList = A(1:middle);
    rightList = A((middle+1):length(A));
    leftList = sort(leftList);
    rightList = sort(rightList);
    A = merge(leftList, rightList);
end
```