

# MGL7460 : Laboratoire #6

## Ajout de commandes au *gem* *minised* et *refactoring*

### Solution

Extraits du fichier lib/minised/minised.rb

```
# a.
def self.print( flux_in, flux_out, motif )
  flux_in.each do |ligne|
    flux_out << ligne if /#{motif}/ =~ ligne
    flux_out << ligne
  end
end

# c.
def self.sub( flux_in, flux_out, motif,
              motif_a_substituer, chaine_de_replacement, global )
  motif_a_substituer_er = /#{motif_a_substituer}/

  flux_in.each do |ligne|
    if /#{motif}/ =~ ligne
      flux_out << (ligne.send (global ? :gsub : :sub),
                    motif_a_substituer_er, chaine_de_replacement)
    else
      flux_out << ligne
    end
  end
end
```

## Extraits du fichier bin/minised avant *refactoring!*

```
# b.
desc 'Imprime les lignes qui matchent un motif'
arg_name 'motif [fichier...]'
command :print do |c|
  c.action do |global_options, options, args|
    motif, *fichiers = args
    fichiers = [:stdin] if fichiers.empty?

    fichiers.each do |fichier|
      MiniSed.traiter_fichier( fichier, in_place: global_options[:i] ) do
        MiniSed.print( flux_in, flux_out, motif )
      end
    end
  end
end

# d.
desc 'Imprime les lignes qui matchent un motif'
arg_name 'motif [fichier...]'
desc "Substitue un motif par une chaine dans les lignes qui matchent"
arg_name 'motif motif_a_substituer chaine_de_replacement [fichier...]'
command :sub do |c|
  c.desc 'Substitution globale'
  c.switch :g

  c.action do |global_options, options, args|
    motif, motif_a_substituer, chaine_de_replacement, *fichiers = args
    fichiers = [:stdin] if fichiers.empty?

    fichiers.each do |fichier|
      MiniSed.traiter_fichier( fichier, in_place: global_options[:i] ) do
        MiniSed.sub( flux_in, flux_out, motif, motif_a_substituer,
                    chaine_de_replacement, options[:g] )
      end
    end
  end
end
```

Extraits du fichier bin/minised après le *refactoring* présenté en classe...  
pour rendre le code plus *DRY*, en utilisant la méthode `commande_sans_option`

```
# e.
def commande_sans_option commande, nb_arguments: 1
  command commande do |c|
    c.action do |global_options, options, args|
      les_args = args.shift(nb_arguments)
      fichiers = args.empty? ? [:stdin] : args
      in_place = global_options[:'in-place']

      fichiers.each do |fichier|
        MiniSed.traiter_fichier(fichier, in_place: in_place) do |flux_in, flux_out|
          MiniSed.send commande, flux_in, flux_out, *les_args
        end
      end
    end
  end
end

desc 'Supprime les lignes qui matchent un motif'
arg_name 'motif [fichier...]'
commande_sans_option :delete

desc 'Imprime les lignes qui matchent un motif'
arg_name 'motif [fichier...]'
commande_sans_option :print
```