VI TEXT EDITOR BRIEF START

1) **vi/vim** (slightly modernized vi) is a common text editor in majority of UNIX distributions, it is fully configurable and useful – it needs only a few commands to know to start editing:

```
    #touch someTextFile.txt && vim someTextFile.txt
    Esc – starting base command mode
    a – insert text ( append mode )
    Esc – return to command ( normal ) mode
    :w – save document
    :q – close document without saving
    :q! - force closing document without saving changes
    :wq – save and close
    :%!xxd - convert current work into hex edition format (no updates on string interpreter)
    :%!xxd -g4 - group bytes into 4B words
    :%!xxd -r - convert back into ordinary file - must be done, before save (file write)
    Ctr + p – word/variables auto filling
    Ctr+o :w – save command :w-save for saving by shortcut Ctr+o during insert mode
    v – visual mode ( highline text fragment ):
        y – copy highlined
        p – paste highlined after cursor
    Esc – return to command ( normal mode )
        yy – copy current line
        yiw – copy current word
        dd – delete current line
        p – paste copied content
    u – undo last change ( 1 change in vi; more in vim )
    Ctr+r/. – repo last change
    ?keyword – find keyword
    :lineNumber – go to file line
    #system-wide clipboard for copying between vi/vim sessions:
        #sudo apt-get install -y vim-gtk
        #copy in  Visual mode:
            "+y
        #paste in other terminal with different vim session:
            "+p
```

2) vim configuration file (add after line „syntax on”):

```
    #sudo vi /etc/vim/vimrc
          set autoindent
          set ts=4
          set sw=4
          set mouse=a
          set number
          set background=dark
          color evening
        map <F5> :w <bar> exec
        \ '!clear && g++ ' .shellescape('%').
        \ ' -o ' .shellescape('%:r').
        \ ' && ./' .shellescape('%:r') <CR>
```

It sets line autoindent, tab space number, block autoindent, mouse handling and line numbers.

3) view – vi file viewer

4) there are other editors, with different controls like **nano**.

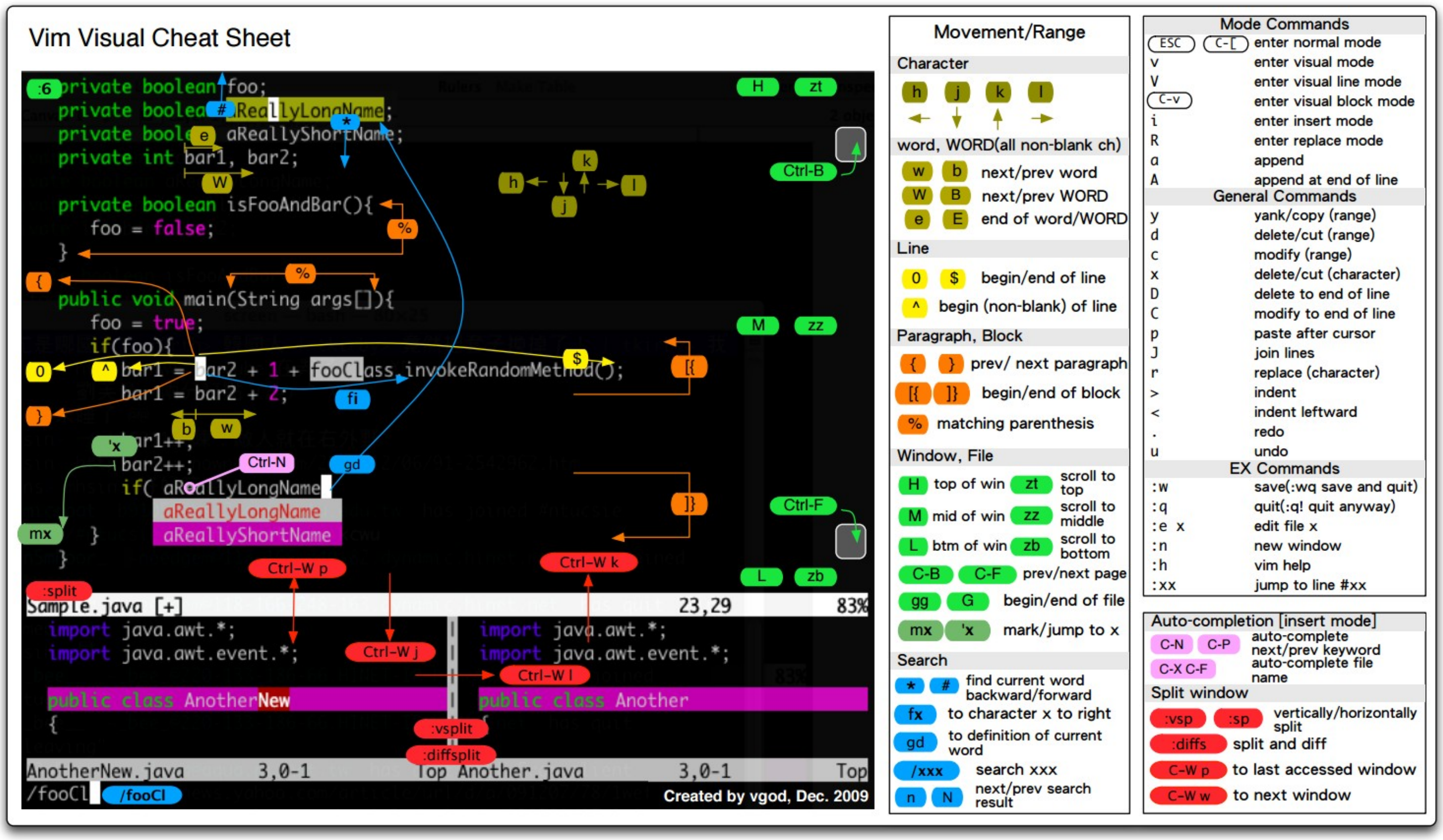
5) terminal colorization – edit file in home directory ~/bashrc:
uncomment line:

```
    #force_color_prompt=yes
```

6) different stuff:

```
    split window
        :sp
        :vsp
```

split window, and open file
    :sp file.txt
switch view to neighbour window:
    Ctr+w arrow
run shell command:
    :! ls
    :! ./RUN_COMMANDS.sh
ctags (does not correctly handle with enums):
    #sudo aptitude install -y exuberant-ctags
        :! ctags -f `pwd`/tags -R *
        :Ctr+]   #jump to the tag
        :tn   #go to next tag definition
        :tp   #go to previous tag definition
        :ts   #list all tag definitions
        :Ctr+T   #jump back in tag stack
cscope:
    #sudo aptitude install cscope
    https://cscope.sourceforge.net/cscope_vim_tutorial.html
    add cscope_maps.vim to .vimrc via added line „source ~/cscope_maps.vim"
        Ctrl+\+s – show tag list within files
        Ctrl+] - go to next tag from tag list
        Ctrl+T – go backward
        vim -t main
        :cscope find symbol main   = :cs f s main
        Ctrl+spacebar+s   - search and open in split horizontally
            Ctrl+W+left/right – jump to another view


Vim Visual Cheat Sheet

```
private boolean foo;
private boolean aReallyLongName;
private boolean aReallyShortName;
private int bar1, bar2;

private boolean isFooAndBar(){
    foo = false;
}

public void main(String args[]){
    foo = true;
    if(foo){
        bar1 = bar2 + 1 + fooClass.invokeRandomMethod();
        bar1 = bar2 + 2;
        bar1++;
        bar2++;
        if( aReallyLongName
                aReallyLongName
                aReallyShortName
    }
}
Sample.java [+]                                  23,29          83%
  import java.awt.*;                 |  import java.awt.*;
  import java.awt.event.*;           |  import java.awt.event.*;

public class AnotherNew              |public class Another
  {
AnotherNew.java     3,0-1        Top  Another.java       3,0-1        Top
/fooCl
```

Created by vgod, Dec. 2009

Movement/Range

Character	
h j k l	← ↓ ↑ →
word, WORD(all non-blank ch)	
w b	next/prev word
W B	next/prev WORD
e E	end of word/WORD
Line	
0 $	begin/end of line
^	begin (non-blank) of line
Paragraph, Block	
{ }	prev/ next paragraph
[{]}	begin/end of block
%	matching parenthesis
Window, File	
H	top of win
zt	scroll to top
M	mid of win
zz	scroll to middle
L	btm of win
zb	scroll to bottom
C-B C-F	prev/next page
gg G	begin/end of file
mx 'x	mark/jump to x
Search	
* #	find current word backward/forward
fx	to character x to right
gd	to definition of current word
/xxx	search xxx
n N	next/prev search result

Mode Commands	
ESC C-[	enter normal mode
v	enter visual mode
V	enter visual line mode
C-v	enter visual block mode
i	enter insert mode
R	enter replace mode
a	append
A	append at end of line
General Commands	
y	yank/copy (range)
d	delete/cut (range)
c	modify (range)
x	delete/cut (character)
D	delete to end of line
C	modify to end of line
p	paste after cursor
J	join lines
r	replace (character)
>	indent
<	indent leftward
.	redo
u	undo
EX Commands	
:w	save(:wq save and quit)
:q	quit(:q! quit anyway)
:e x	edit file x
:n	new window
:h	vim help
:xx	jump to line #xx

Auto-completion [insert mode]	
C-N C-P	auto-complete next/prev keyword
C-X C-F	auto-complete file name
Split window	
:vsp :sp	vertically/horizontally split
:diffs	split and diff
C-W p	to last accessed window
C-W w	to next window

shortcut:
    :nnoremap <keystroke> vimCommand