/**

Pattern p=Pattern.compile("[abc]"); // pattern is a compile version of Regular expression of the object passed and to create the object use compile()

Matcher m=p.matcher ("abbbaabbba");

in pattern argument passed below in "[ ]"

1. [abc] -> either a or b or c
2. [^abc] -> except a.b.c all the letters Note: '^' is negation symbol that is not this
3. [a-z] -> any lower case symbol from a to z
4. [A-Z] -> any upper case symbol from A to Z
5. [a-zA-z] -> all the alphabet symbols
6. [0-9] -> any digit from 0 - 9
7. [0-9a-zA-Z] -> any alphanumeric symbol
8. [^0-9a-zA-Z] -> any symbol except this characters i.e all special characters

**********above are know as character classes **********

Example:

String x=[];

Pattern p=pattern.compile(x);

Matcher m=p.matcher("a 3 b # k @ 9 z");

0 1 2 3 4 5 6 7

x=[abc] // o/p = 0...a 2...b

x=[^abc] // o/p = 1...3 / 3...# / 4...k / 5...@ / 6...9 / 7...z

x=[a-z] // p/p = 0...a / 0...b / 4...k / 7...z

x=[0-9] // o/p= 1...3 / 6...9 /

x-[a-bA-Z0-9] // o/p= 0...a / 1...3 / 2...b / 4...k / 6...9 / 7...z

x=[^0-9a-zA-Z] // o/p = 3...# / 5...@

********** predefined classes **********

\s => spcae character

\S => except spcae character

\d => any digit character [0-9]

\D => except digit from [0-9]

\w => any word character [0-9a-zA-Z]

\W => except any word chracter [special symbol]

. => any charchter all symbols

String x=[];

Pattern p=pattern.matcher(x);

Matcher m=p.matcher("a3b k@9z");

01234567

x=\\s => 3...

x=\\S => 0...a 1..3 2..b 4...k 5..@ 6..9 7..z

x=\\d => 2..1 6..9

x=\\D => 0...a 2..b 3... 4...k 5..@ 7..z

x=\\w => 0...a 1..3 2..b 4...k 6..9 7..z

x=\\W => 3.. @..5

x=. => 0...a 1..3 2..b 3... 4...k 5..@ 6..9 7..z

*********Qunatifers***************

we use thtis to specify to number of occurences to mtach

1. a => excatly one a

2. a+ => atlest one a

3. a* =>any number of a's includeong zero number also

4. a? => atmost one number of a

String x=[];

Pattern p=pattern.compile(x);

Matcher m=p.matcher("abaabaaab");

012345678

x=a => 0..a 2..a 3..a 5..a 6..a 7..a

x=a+ => 0..a 2..a 5..a

x=a* => 0..a 1.. 2..aa 4.. 5..aa 8... 9...(it searchs for end +1 position)

x=a? => 0..a 1.. 2..a 3..a 4.. 5..a 6..a 7..a 8.. 9..(it searches for end +1 position)

**********split method**********

Pattern p=Pattern.compile("\\s");

String[] s=p.split("mayuresh sunil Zende")

for(String s1:s)

System.out.println(s1);//mayuresh \n sunil \n zendes

2) Pattern p=Pattern.complie("s"); => mayure \n h \n unil zende

3) Pattern p=Pattern.compile("\\."); oer we can use ("[.]")

tring[] s=p.split("www.mayuresh.com");

for(String s1:s)

System.out.println(s1); // o/p we will get the string seprated by . i.e www \n mayureh \n zende

*******to validate mobile number ****(10 digit)

Re is => [7-9][0-9]{9} // i.e first number is 7/8/9 anf next 9 diggit can be anything

for 11 digit we can add 0?[7-9][0-9]

for 12 digit we can add (0/91)?[7-9][0-9]

**********for mail iD we can use the**********

Re [a-zA-Z0-9][a-zA-Z0-9_.]*@[a-zA-Z0-9]+ ([.][a-zA-Z+])+

----------for sppecific gmai use ----------

Re [a-zA-Z0-9][0-9a-zA-z._]* @gmail[.]com

****for language identifer **********

rules

1) a-k

2) [0369] //divsible by 3

3) {a-zA-Z0-9$#}

RE [a-k][0369][a-zA-Z$#]*

*/