**Cytogenetics­­­**

Algorithm for how to check the formula: -

**Take the input to analyses if the input corrects or not**

1. check if the input empty or not => if the input is empty throw exception ("The ISCN Formula must not be an empty string.")

* after making sure that input is not empty it’s start some validation in general to set some flag value some of this condition may case to **throw an exception**
* first thing is to clear some use useless char by replace each ( **" ".** ) to non and replace each x to capital X
* then separate the string in to array for items separated in each (**,**)
* and starting from last array value start some validation

**this condition for last array value**

1. If the string end with [cp] or [number**cp**] or [number]   
   - if this condition was cp then it set **mbooCompositeKaryotype to** True **-** if the [] contain number mnCloneSize = to number value  
   - then it spilt the much part from the sting

**EX:** [cp] [cp10] [10]; it must be at the end of the string

1. If the string end with inc   
   - if the condition was true it set mbooIncompleteKaryotype = True  
   - then it spilt the much part from the sting

**EX:** inc => it should be in the end and not **acceptable** like **9inc**   
 the better thing is like ‘9 inc’ or 9)inc

1. If the string end with [inc]  
   - if the condition was true it set mbooIncompleteKaryotype = True  
   - then it spilt the much part from the sting

**Note: the app set in comment this condition not officially not correct!**

1. Check if the string contains dmin **Case sensitive**   
   if it true - {// still not implemented}
2. Check if the string contains mar **IgnoreCase**   
   if it true - {// still not implemented}
3. Check if the string is **ring**   
   what I mean: value r value and it depend on before the R on after it

* Before r it accepts (+ - ~ any white space, any number)  
  letter r   
  after r it accepts (- [] ~ c x ? any white space any number)
* examples**:** +9r[10]x  
  if it true - { // still not implemented }

**now the condition starts from first item of array string**

1. check if the first string before (**,**) in correct format like **number or number< ploidy n >**

condition description   
 - first letter can be (? - ~) **optional**  
 - then it the number without any space between this character and number

- between numbers it accepts some character (- ~) and again without any space => **optional**  
- then it can accept some special character like (‘< or (‘) and inside it accepts number and this number should be written with n litter in capital or small => **optional** the character should close If not it gets error (‘> or)’)

**There is an error in this case as the app can accept this shape 45<2n) can’t define the close character**   
EX: (?, ?45, ~45-47, 45<2n>, -48-50<2n>)

If the condition was falseit get an exception **=>** Non-specified error in chromosome count element

If the number is zero it gets an exception => Error in chromosome count element (" & formula "): Cannot determine number of chromosomes.

If the condition was true {

* If there an **–** between number it set mbApproximate to true
* It calculates Ploidy by different ways

1. If there a number before n it takes that number be default as **Ploidy**
2. If there **–** between numbers it sums two numbers divided by 46
3. If it only one number it sums this number twice divided by 46

**Note that: -** the Ploidy is the value to the nearest integer **=> 1.5 is 2 / 1.4 is 1**

}

**now the condition starts from second item of array string this condition to determine if there** noSexChromosomes or not

* First it checks if the string contains character like x or y or c   
  if it not contains any of these it set noSexChromosomes like abc in this case it set this flag to true   
  **else it set** noSexChromosomes to false and it start to calculate the number of each x,y,c and number of question for each x,y,c

now it starts condition from the item after SexChromosomes and if the string by default not contain SexChromosomes it start from the item after the number

* First it starts a loop of the string after the number or SexChromosomes in this loop it checks if their word **or** if it false it will not throw an error it will continue  
  if it true it split **or** and replace it with **?** andif the string after split **OR** start with + or – it remove this characters
* Then it check if the input contain any not allowed character like(#,$,@,……,)  
  if the app contain some of these character it get exception invalid character

only digits, "normal" characters, +-x~()[]; are acceptable

from there the app will start to determine in some different condition which type he deal with and this type are

a) t Translocation

b) del Deletion

c) add Addition

d) dup Duplication

e) dic Dicentric Chromosomes

f) i Isochromosome

g) idic Iso-dicentric Chromosomes

h) ins Insertion

i) inv Invertion

j) r Ring Chromosome

k) trp Triplication

l) idem repetition of previous aberrations

m) tas telomeric associations

n) hsr homogeneously staining regions

o) trc tricentric chromosome

p) qdp quadruplication