**020\*1(132+223\*4(325+426) 7+528+639+7410)**

|  |  |  |
| --- | --- | --- |
| **Действие** | **Метод, описание способа, вспомогательные методы** | **Результат** |
| Ιесли есть скобки в скобках | If (contains)` | True |
| 1находим конечную скобку блока | Все делаем до того как определить конечную скобку внутреннего блока | **10** |
| -считаем все открывающие скобки до первой закрывающей – получаем кол-во скобок внутри блока | While (indx+i++!=’)’{  If (indx+i ==’)’{ count ++}  } count = кол-во открывающих скобок |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| 2считаем сколько цифр нужно перенести между скобками, одновременно создавая массив индексов знаков которые нужно будет перенести | While(indx = 10-x++ != ‘)’  Queue <Integer>.add(indx) | X = кол-во цифр между скобками которые надо перенести  Queue offsetIndxSigns список индексов знаков которые нужно перенести |
| 3. Создаем саблист переносимых знаков |  | sublistOffsetSigns |
| 4.создаем массив индексов цифр которые нужно перенести (семещение -2) | Queue<Integer> makeIndxOfCiphers | Queue offsetIndxCiphers |
| 5.Создаем саблист переносимых цифр | От первого значения из offsetIndxCiphers до последнего+1 | sublistOffsetCiphers |
| 6.определяем индекс первой цифры блока ( он равен индексу начальной скобки блока | Loparenth=loCipher |  |
| 7.добавляем саблист Цифр на этот индекс | Chiphers.addAll(loCipher, sublistOffsetCiphers) |  |
| 8. удаляем цифры со старого места по индексам из offsetIndxCiphers |  | Цифры перенесены |
| 9 добавляем саблист Знаков на индекс Loparenth+1) | Signs.addAll(Loparenth+1, sublistOffsetSigns) |  |
| 10.удаляем знаки со старого места по индексам из sublistOffsetSigns |  | Знаки перенесены  2\*(2+3+4+3+2\*(2+2)) |
|  |  |  |