

***Relatório ESINF***

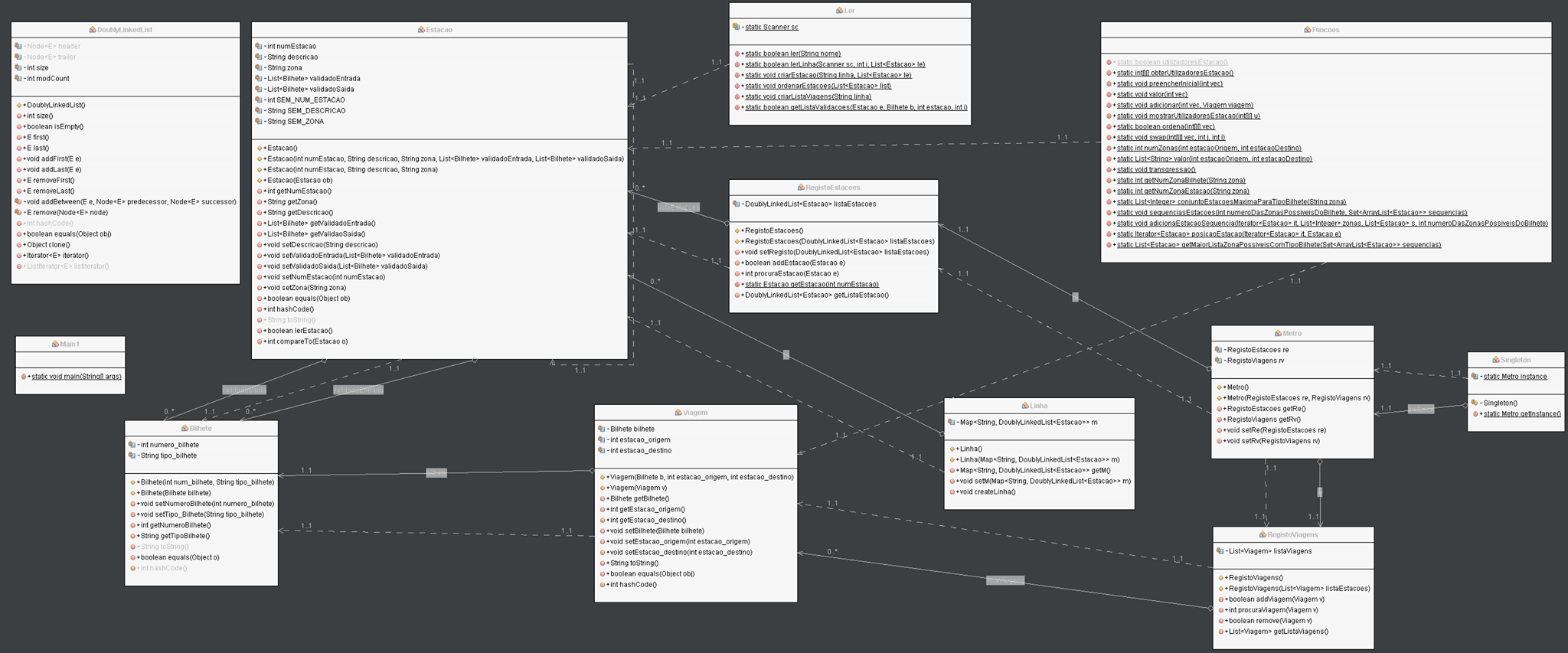
Turma 2Dk:

JoãoFlores(1171409); José Mota (1161263);

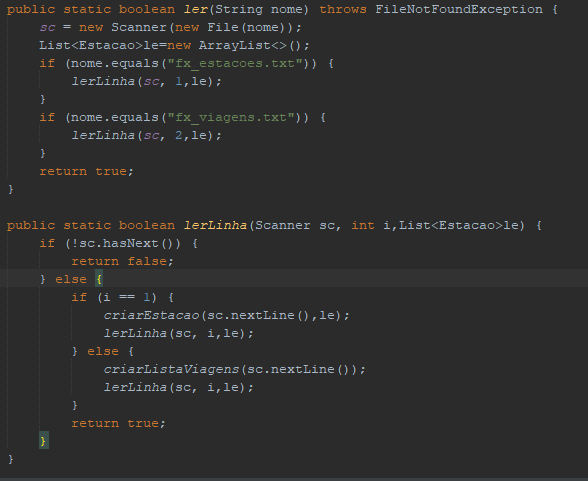
**Metro Service Project**

Pretende-se desenvolver um conjunto de classes e respetivos métodos que permitam gerir a utilização de uma linha de metro. Os utilizadores podem usar o serviço se possuírem um bilhete. Em cada estação de metro existe uma máquina que permite ao utilizador validar o bilhete. O utilizador faz sempre a validação do bilhete na estação origem e na estação destino. A linha de metro possui várias zonas e cada zona abrange um conjunto de estações.

Para o desenvolvimento desta primeira parte do trabalho, criamos as seguintes classes: Bilhete, Estacao, Ler, Linha, Metro, RegistoEstacoes, RegistoViagens e a classe Viagem. No diagrama seguinte encontram-se as classes usadas, a sua relação em si bem como atributos e métodos de cada uma.

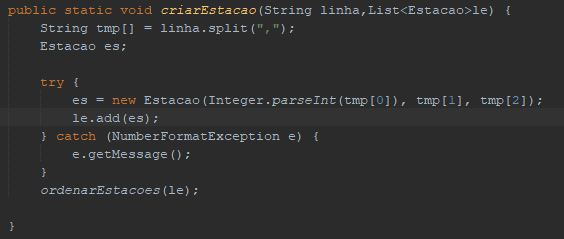


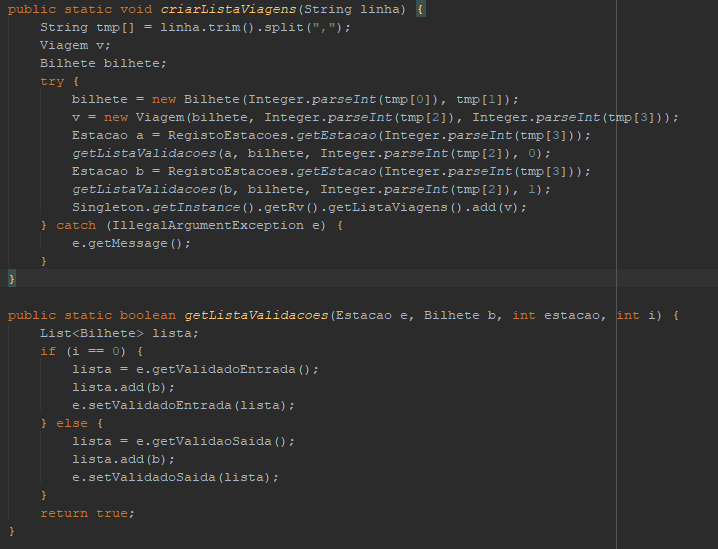
**1) Leitura de ficheiros:**

Para o programa ter funcionamento é necessário que se importe dados. Lendo linha a linha os ficheiros fornecidos.

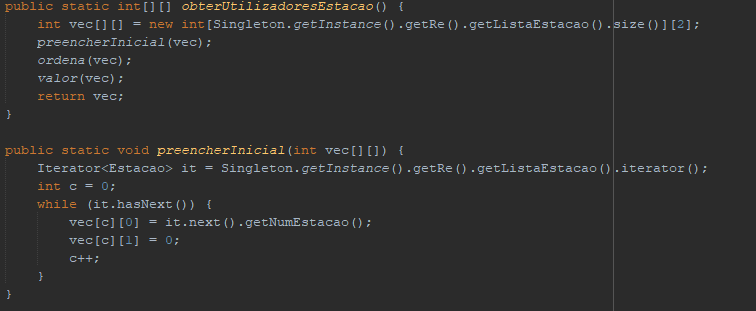
Começamos pro criar um scanner para ler o ficheiro passado por parametro como String, no metodo ler linha cria-se um certo objeto para cada linha lida.

No caso da estação separa-se a linha por “,” pois no ficheiro cada atributo encontra-se separado por isso. De seguida cria-se a estação com os parametros pela ordem no construtor.



No caso das viagens o split tem o mesmo sentido. A diferença é que se adiciona as estações no onde começou a viagem e onde terminou.

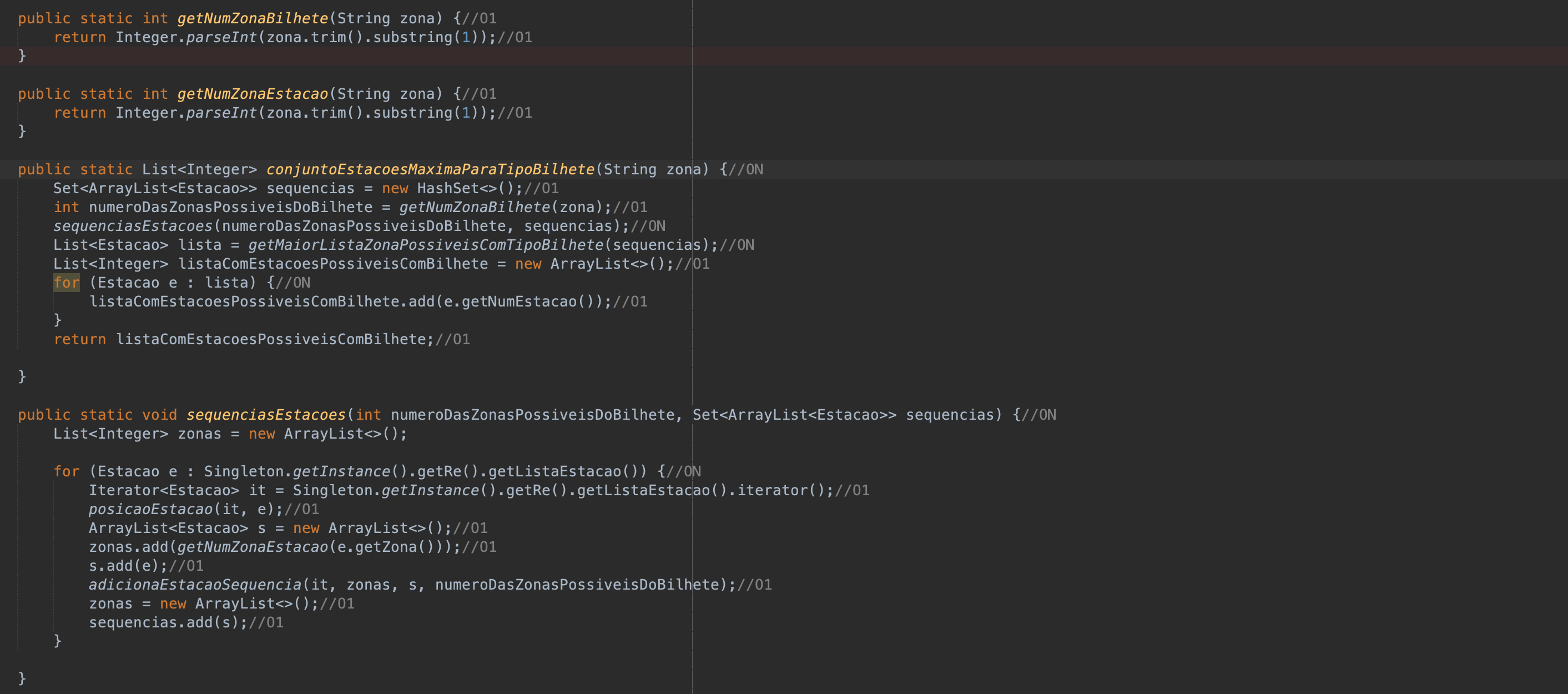
**2) Calcular a quantidades de utilizador que passam por cada estação:**

Para obter os utilizadores instancia-mos uma matriz de dimensão igual ao número de estações e 2. Para além disso iniciamos os valores todos da matriz na possição [x][1]=0 pois sendo um contador tem de estar a 0. A primeira possição vai indicar a estação a segunda vão ser referente ao contador.

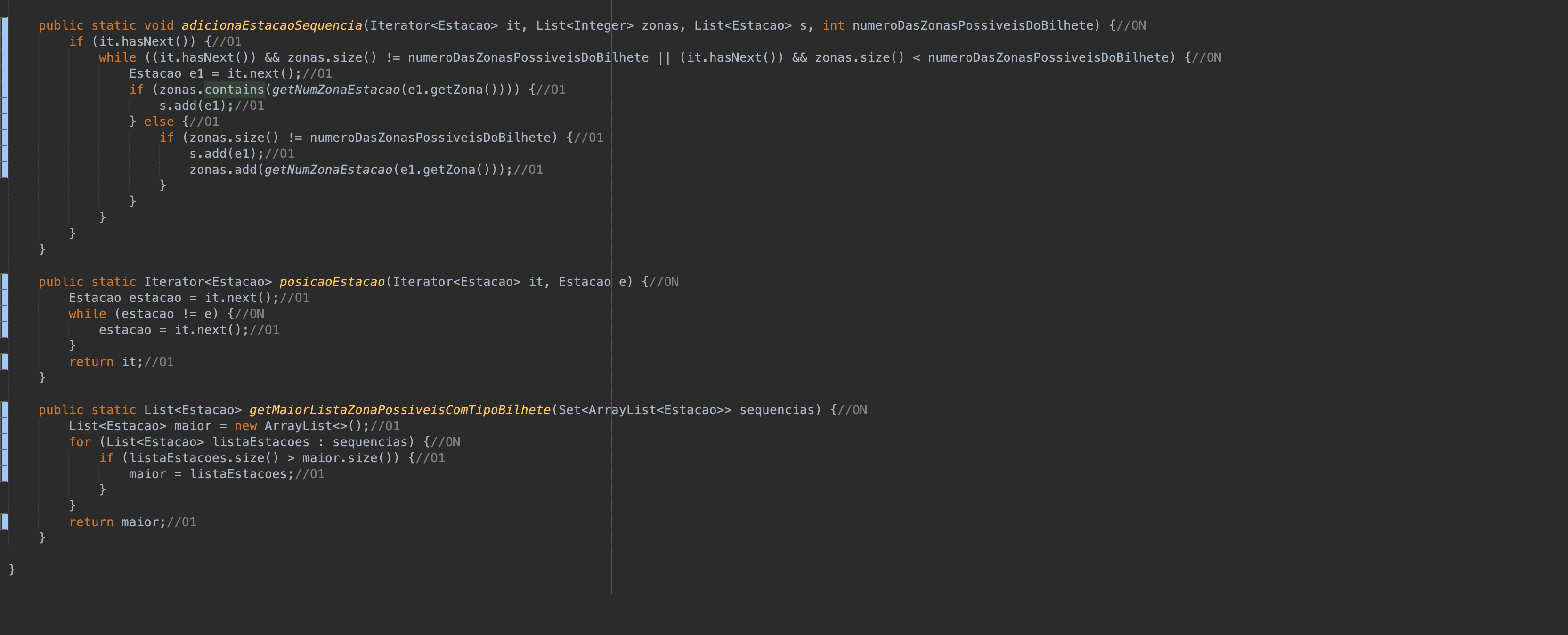
De seguida percoremos cada viagem realizada e fomos adicionar a matriz cada passagem, sendo cada passagem iniciada pela estacao de origem e indo até à estação de destino daí a utilização do ciclo for para incrementar um a cada estação que foi passada naquela viagem. A notação de “k-1” é devido ao factor de as estações começarem na número 1 e o index da matriz no 0.

**3) Calcular a maior sequencia de estações que é possivel percorrer com um certo bilhete:**

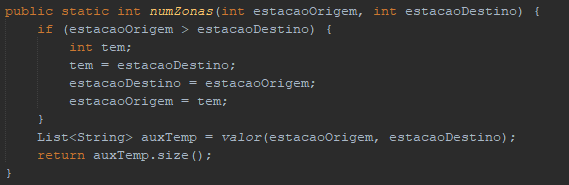
Para determinar a maior sequência de estações que podemos percorrer com um determinando bilhete, precisamos de saber que tipo de bilhete é ou seja quantas zonas podemos percorrer com um dado bilhete, bem como a zona de cada estacão.

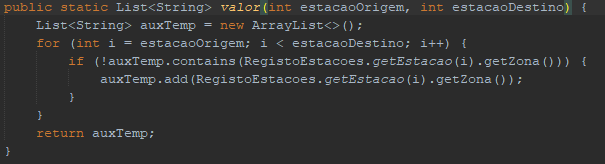
Para tal, retiramos a parte numérica da String zona quer referente ao bilhete quer a zona da estacao(2 primeiros métodos).No método seqeunciasEstacoes vamos através de um ciclo for e de um iterador as estacoes que se encontram na DolublyLinkedList para passarmos o numero da estacao para uma List de Integers.

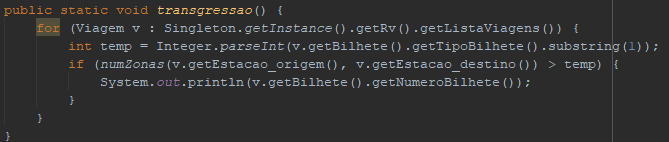
No método adicionarEstacaoSequencia, adicionamos o numero da estação a uma List (zonas), bem como adicionamos a estacão a uma lista que contem estações. No método procurar estação procuramos a estacão pretendida .

Por fim, como queremos a maior lista de sequencias de estacoes, vamos no metodo getMaiorListaZonaPossivelComTipoBilhete, ver qual a maior lista que temos e retornamos a mesma.

**4) Obter os bilhetes em transgressão:**

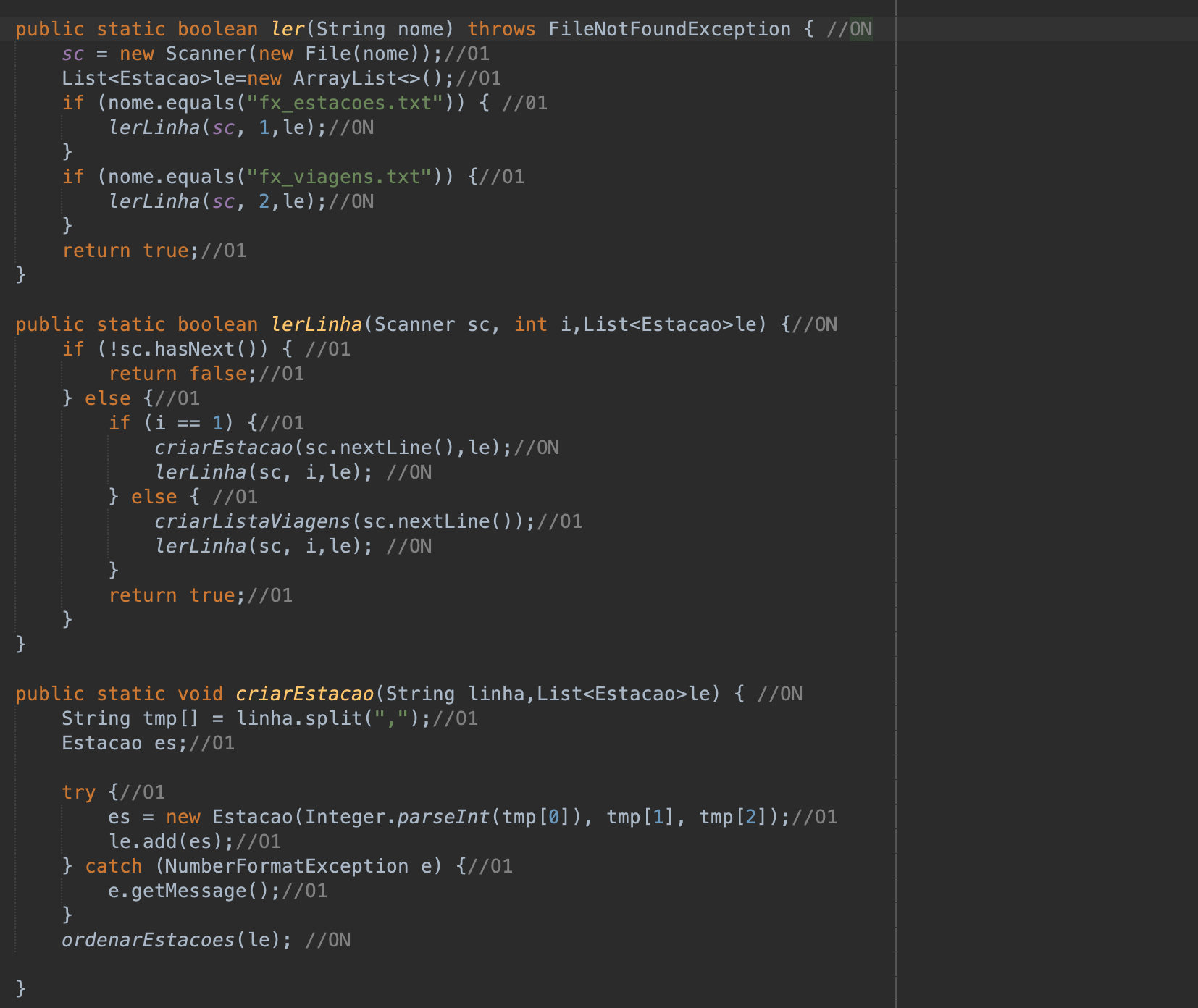
Para um bilhete estar em transgressão é necessário que este passe por mais zonas do que pode, por exemplo comprando um z2 so pode passar por 2 zonas. Para verificar se o bilhete está em transgressão verifica-se a estação de origem e a de destino obtendo as respetivas zonas.

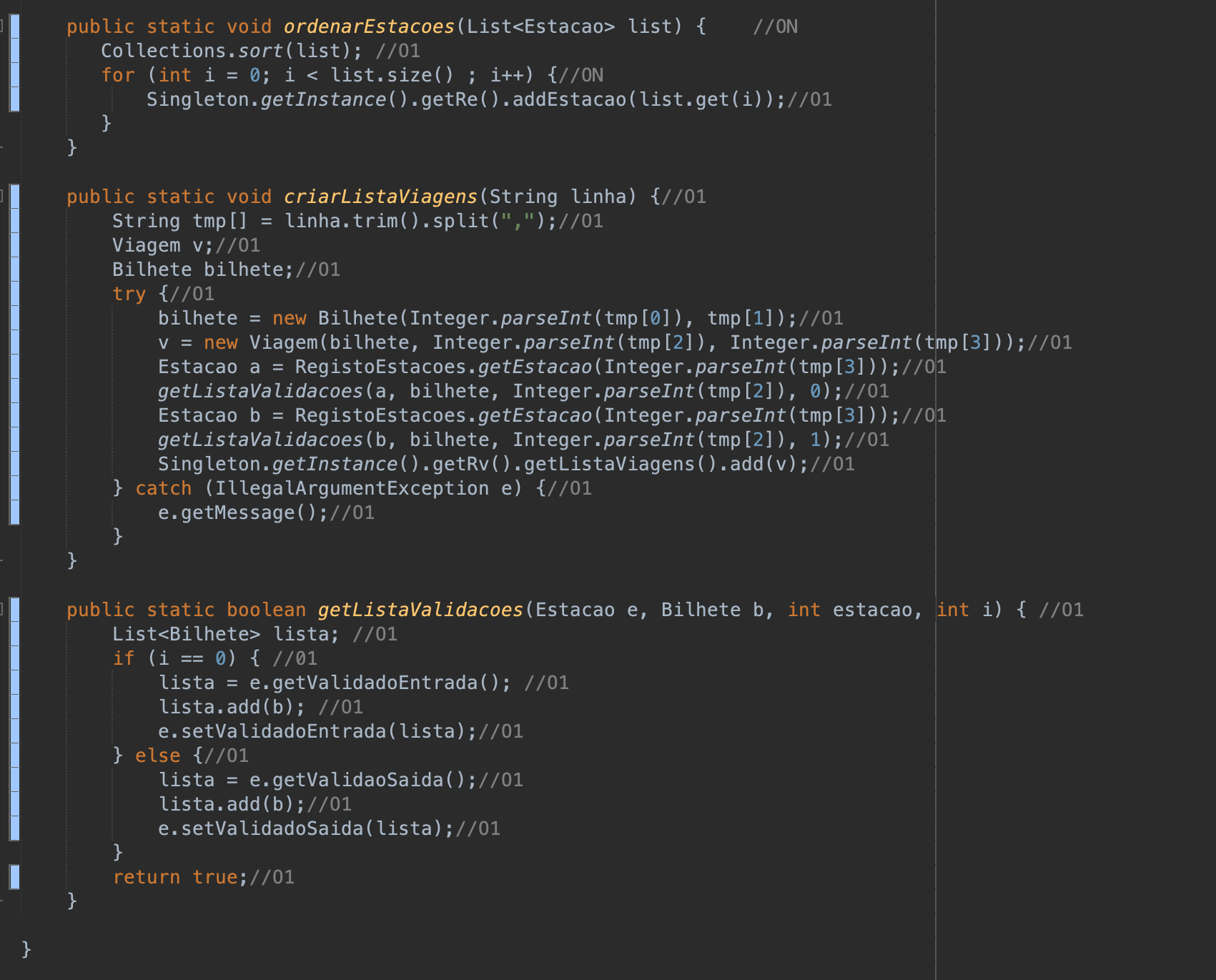
 Após obter as estações é necessário obter cada zona das estações que o utlizador passou. Sendo adicionada a uma lista sempre que for encontrada uma estação que ainda nao tinha sido introduzida.

Finalmente para verificar se foi realizada alguma anomalia verifica-se se o número de zonas que o bilhete podia passar é menor ou igual ao número de zonas que passou caso isso nao aconteça é dado o “alerta”.

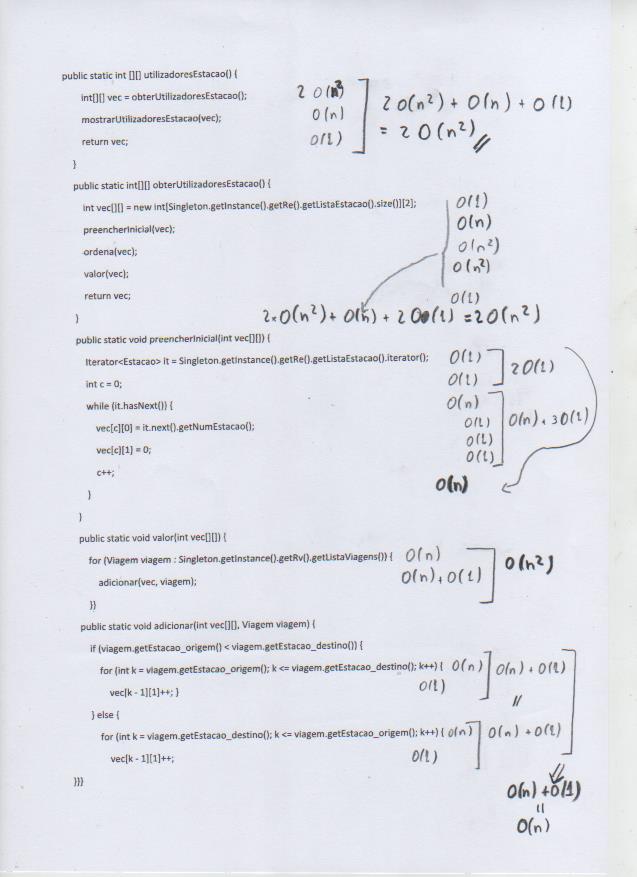
**ANALISE DE COMPLEXIDADE**

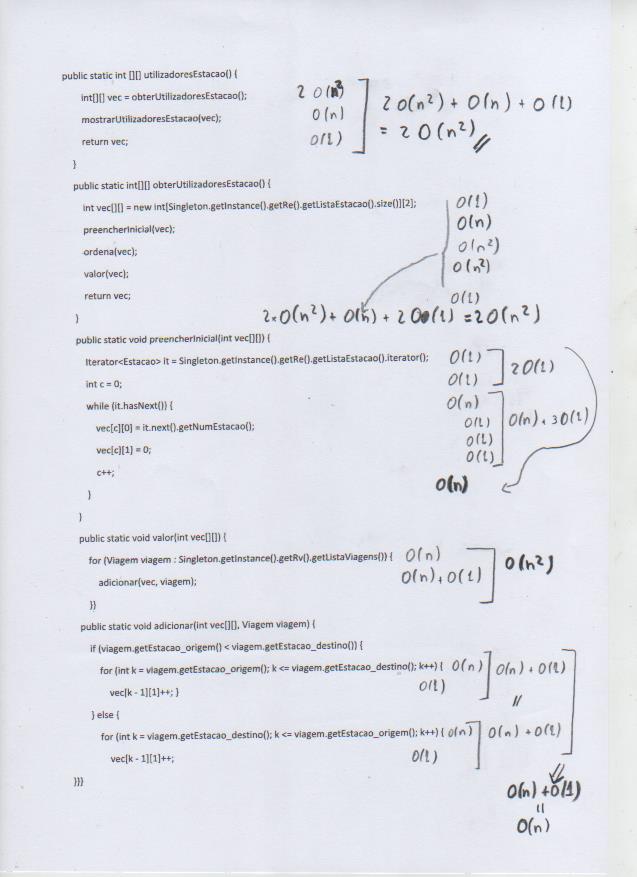


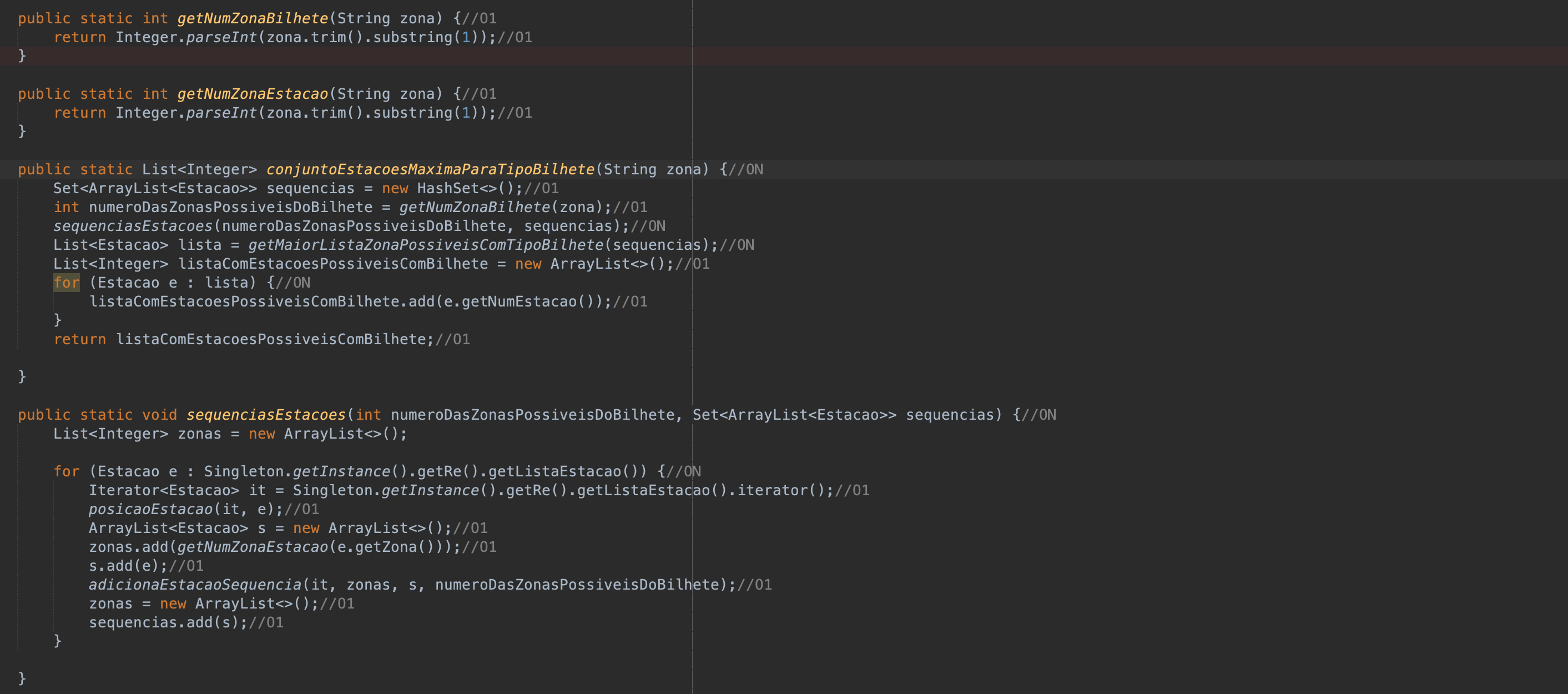


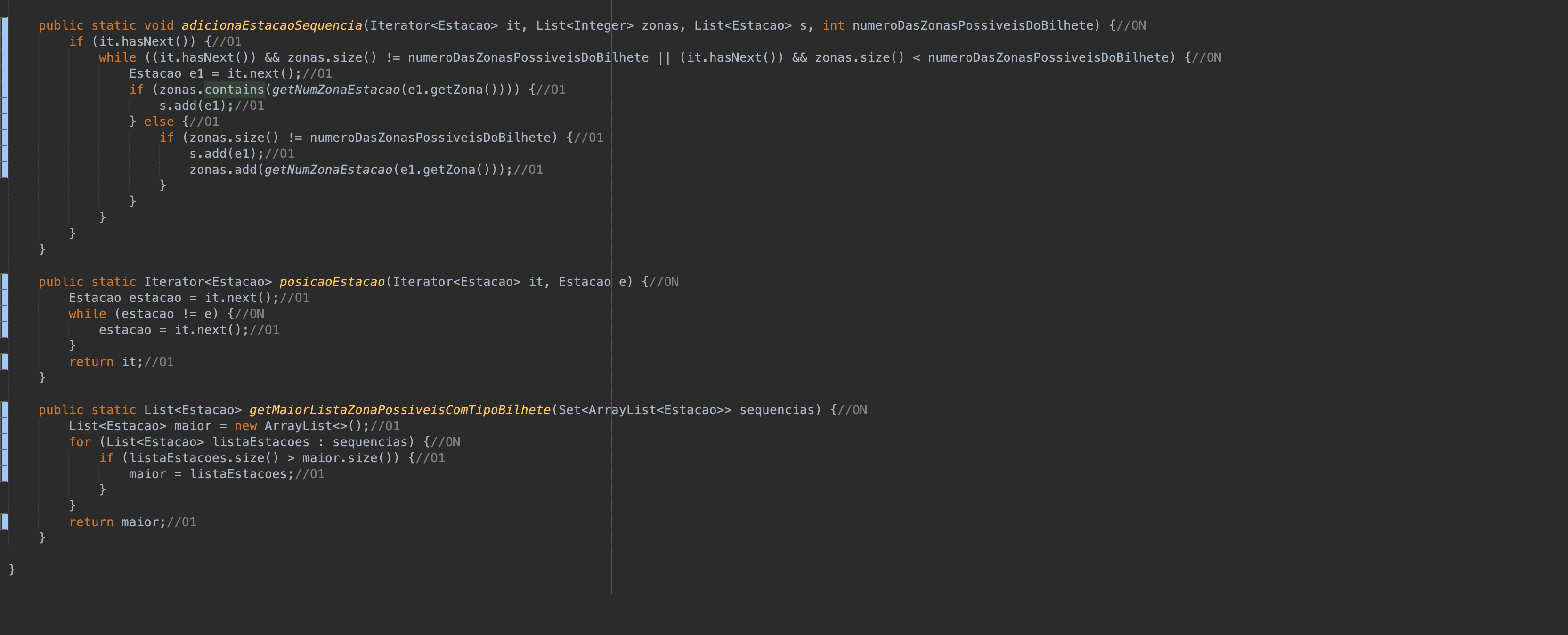
****

**2)**

****

**3)**

**4)**

****