Exercício 1:

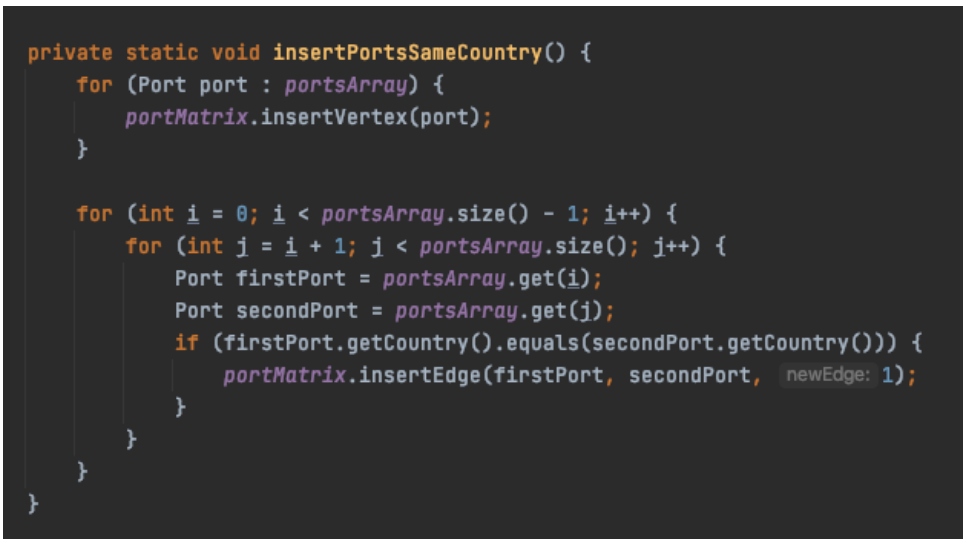
Esta função insere e verifica se a capital de um país tem ligação direta com as capitais dos países com os quais faz fronteira e se sim, insere uma ligação entre as duas capitais. No primeiro for loop criamos os vértices da matriz de adjacência, no segundo loop obtemos as fronteiras e criamos a as arestas entre as capitais dos países.

O(2 * n) + O(log n) = O(n)

```
22
23      private static void insertCapitals() {
24          for (Country country : countriesArray) {
25              capitalMatrix.insertVertex(country.getCapital());
26          }
27
28          for(Border border : borderArray) {
29              String capital1 = border.getCountry1().getCapital();
30              String capital2 = border.getCountry2().getCapital();
31
32              capitalMatrix.insertEdge(capital1, capital2, newEdge: 1);
33          }
34      }
```

Em seguida inserimos Portos em cada vértice na matriz portMatrix e verificamos se são do mesmo país.

O(n) + O(2*n) + O(log n) = O(n)



```
private static void insertPortsSameCountry() {
    for (Port port : portsArray) {
        portMatrix.insertVertex(port);
    }

    for (int i = 0; i < portsArray.size() - 1; i++) {
        for (int j = i + 1; j < portsArray.size(); j++) {
            Port firstPort = portsArray.get(i);
            Port secondPort = portsArray.get(j);
            if (firstPort.getCountry().equals(secondPort.getCountry())) {
                portMatrix.insertEdge(firstPort, secondPort, newEdge: 1);
            }
        }
    }
}
```

Para calcular as distencias dos Portos e obter os N portos mais próximos, usamos a função calculateDistancesFromPorts que tem a complexidade de:

O(n) + O(2*n^2) = O(n)

```
private static void closestPortToCapital() {
    Port nearestPort = null;
    double distance = 0.0;
    for(Country country : countriesArray) {
        for (Port port : portsArray) {
            if(port.getCountry().equals(country.getName())) {
                if (distance == 0.0) {
                    distance = Calculator.getDistance(country.getLatitude(), country.getLongitude(),
                            port.getLatitude(), port.getLongitude());
                    nearestPort = port;
                } else {
                    double distanceToCapital = Calculator.getDistance(country.getLatitude(), country.getLongitude(),
                            port.getLatitude(), port.getLongitude());
                    if (distanceToCapital < distance) {
                        distance = distanceToCapital;
                        nearestPort = port;
                    }
                }
            }
        }
        if (nearestPort != null) {
            portMatrix.insertEdge(nearestPort, nearestPort,  newEdge: 1);
        }
    }
}
```

Nesta função calculamos a distância entre portos de um país, que para além de se conectarem com todos os portos do mesmo país, o Porto mais próximo da capital do país conecta-se a ele próprio.

O(n) + O(2*n^2) = O(n)

```
private static void calculateDistancesFromPorts(int n) {
    ArrayList<PortDistance> distanceArray = null;

    for (Port firstPort : portsArray) {
        distanceArray = new ArrayList<>();
        for (Port secondPort : portsArray) {
            if (!firstPort.getCountry().equals(secondPort.getCountry())) {
                double distanceToPort = Calculator.getDistance(firstPort.getLatitude(), firstPort.getLongitude(),
                        secondPort.getLatitude(), secondPort.getLongitude());
                distanceArray.add(new PortDistance(secondPort, distanceToPort));
            }
        }
        Collections.sort(distanceArray);
        int i = 0;
        for (PortDistance portDistance : distanceArray) {
            if (i < n) {
                portMatrix.insertEdge(firstPort, portDistance.getPort(),  newEdge: 1);
                i++;
            }
        }
    }
}
```

Por fim, o exercício 1 tem a complexidade de O(n).

Autor: Rui Gonçalves, 1191831

Data: 01/01/2022