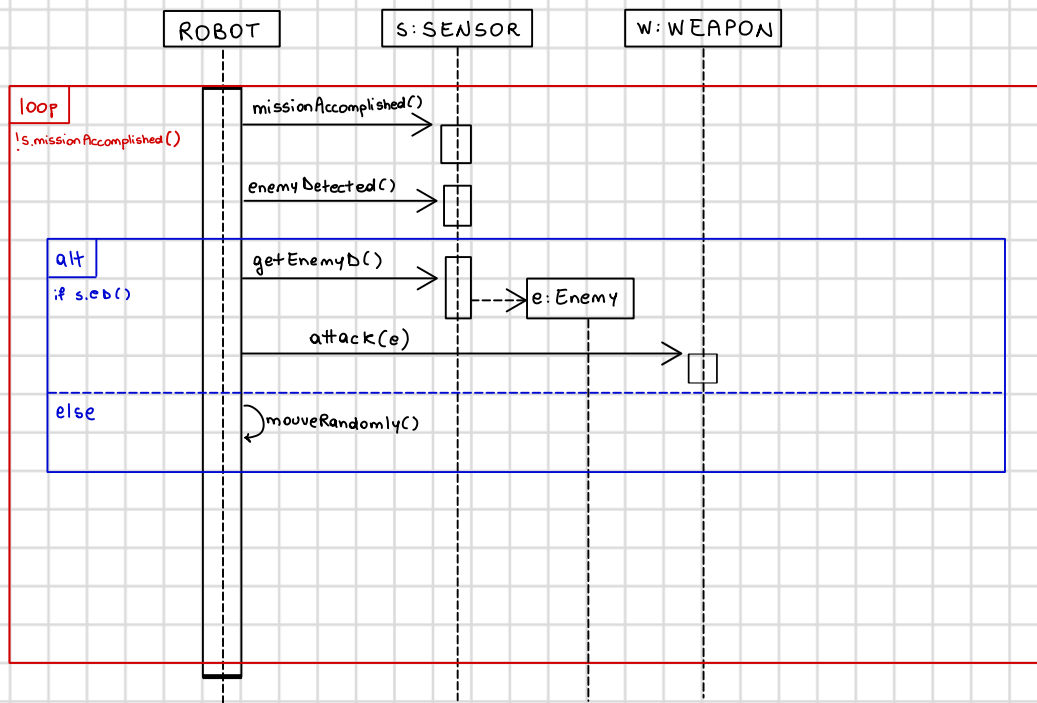


Esercizi su Sequence Diagram

Esercizio 1

Si realizzi un Sequence Diagram relativo a una invocazione del metodo start() della classe Robot riportata di seguito. Si assuma che tutti gli import necessari siano correttamente definiti.

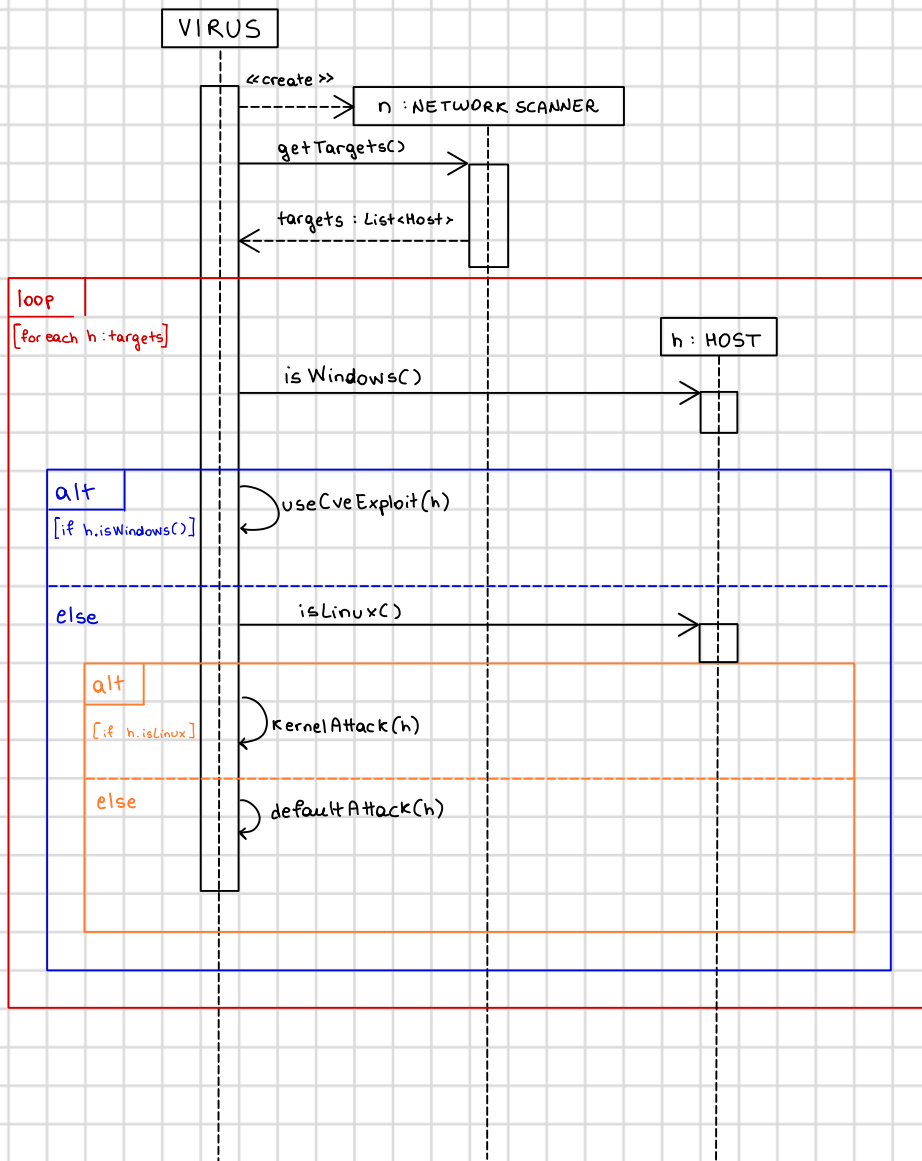
```
public class Robot {
    Sensor s = new Sensor();
    Weapon w = new LaserTurret();
    void start() {
        while(!s.missionAccomplished()){
            if(s.enemyDetected()){
                Enemy e = s.getEnemyDetails();
                w.attack(e);
            } else {
                moveRandomly();
            }
        }
    }
    /* Eventuali altri metodi omissi per brevità */
}
```



Esercizio 2

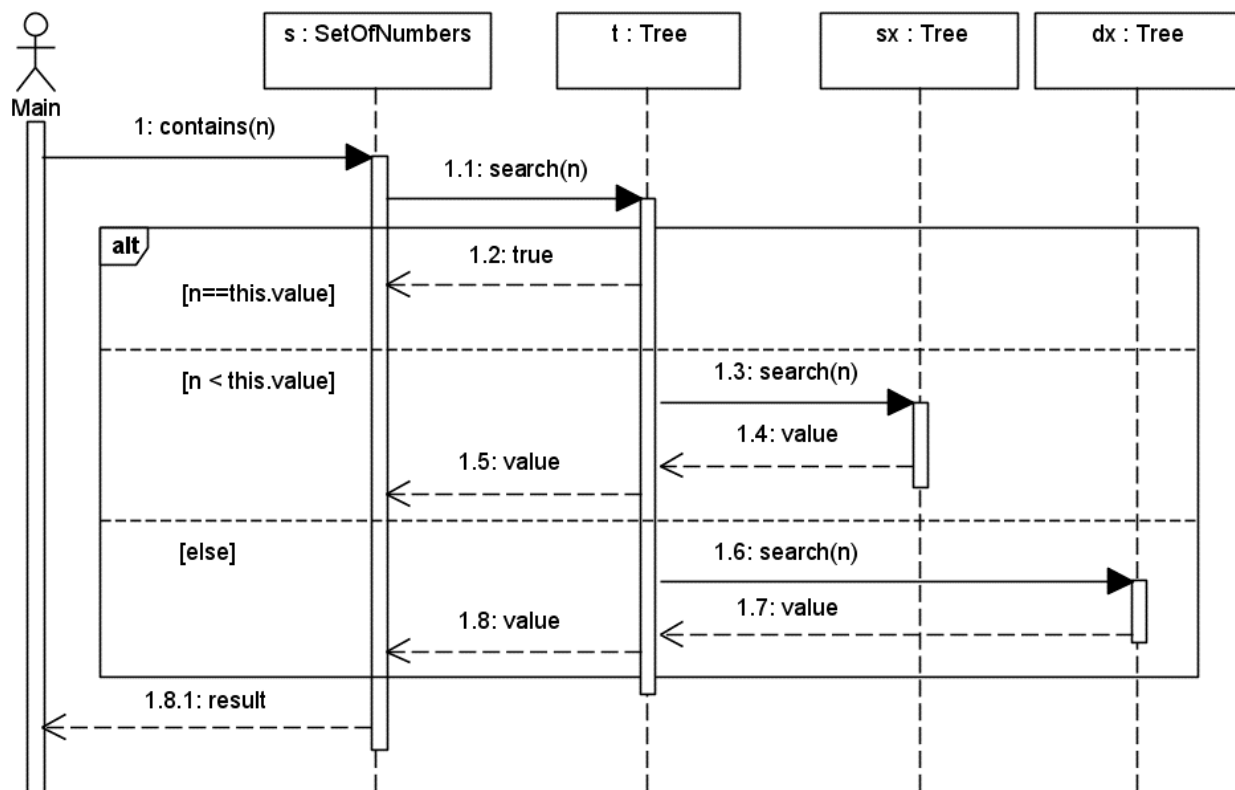
Si realizzi un Sequence Diagram relativo a una invocazione del metodo infect() della classe Virus riportata di seguito. Si assuma che tutti gli import necessari siano correttamente definiti

```
public class Virus {  
    void infect() {  
        NetworkScanner n = new NetworkScanner();  
        List<Host> targets = n.getTargets();  
        for(Host h : targets) {  
            if(h.isWindows()) {  
                useCveExploit(h);  
            }  
            else if(h.isLinux()) {  
                kernelAttack(h);  
            }  
            else {  
                defaultAttack(h);  
            }  
        }  
    }  
  
    /* Eventuali altri metodi omessi per brevità */  
}
```



Esercizio 3

Si scriva tutto il codice Java che è possibile desumere dal seguente Sequence Diagram. Si assuma che la variabile *n* sia di tipo intero.



```
public class Main {
    ...
    SetOfNumbers s;
    ...
    result = s.contains(n);
    ...
}

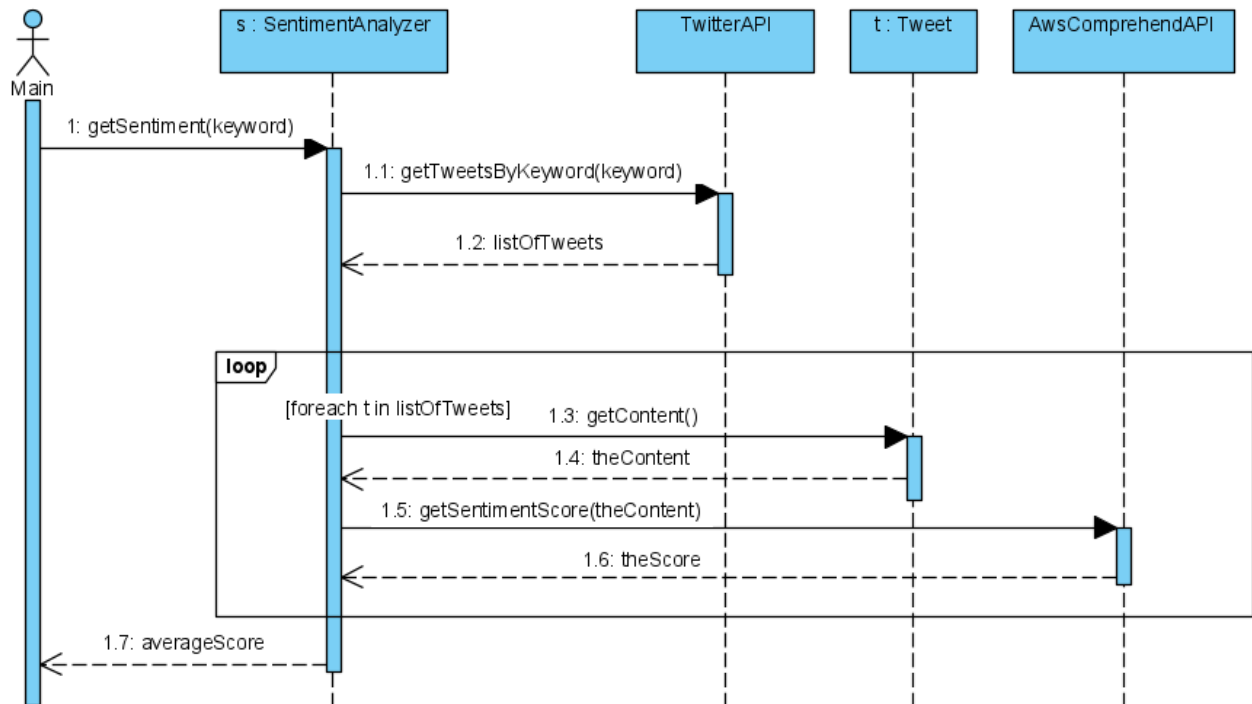
public class SetOfNumbers {
    Tree t;
    public boolean contains(... n) {
        return t.search(n);
    }
}

public class Tree {
    Tree sx, dx;

    public void search(... n) {
        if (n == this.value) {
            return true;
        } else if (n < this.value) {
            return sx.search(n);
        } else {
            return dx.search(n);
        }
    }
}
```

Esercizio 4

Si scriva tutto il codice Java che è possibile desumere dal seguente Sequence Diagram. Si assuma che la variabile `keyword` sia di tipo `String`.



```
1  public class Main {
2      ...
3
4      SentimentAnalyzer s;
5
6      ...
7
8      - averageScore = s.getSentiment(keyword);
9
10     ...
11 }
12
13
14 public class SentimentAnalyzer {
15     ...
16
17     public - getSentiment(... keyword) {
18         ArrayList<Objects> listOfTweets = TwitterAPI.getTweetsByKeyword(keyword);
19     }
20
21     ...
22
23     foreach(Tweet t : listOfTweets) {
24         - theContent = t.getContent();
25         - theScore = AwsComprehendAPI.getSentimentScore(theContent);
26     }
27
28     ...
29
30     return averageScore;
31 }
32
33
34 public class TwitterAPI {
35     public static ArrayList<Object> getTweetsByKeyword(... keyword) {
36         ...
37
38         return listOfTweets;
39     }
40 }
41
42
43 public class Tweet {
44     - content;
45
46     ...
47
48     public - getContent() {
49         return this.content
50     }
51
52     ...
53 }
54
55
56 public class AwsComprehendAPI {
57     ...
58
59     - score;
60
61     public static ... getSentimentScore(... theContent) {
62         ...
63
64         return this.score;
65     }
66 }
```