Class that contains 5 fields:
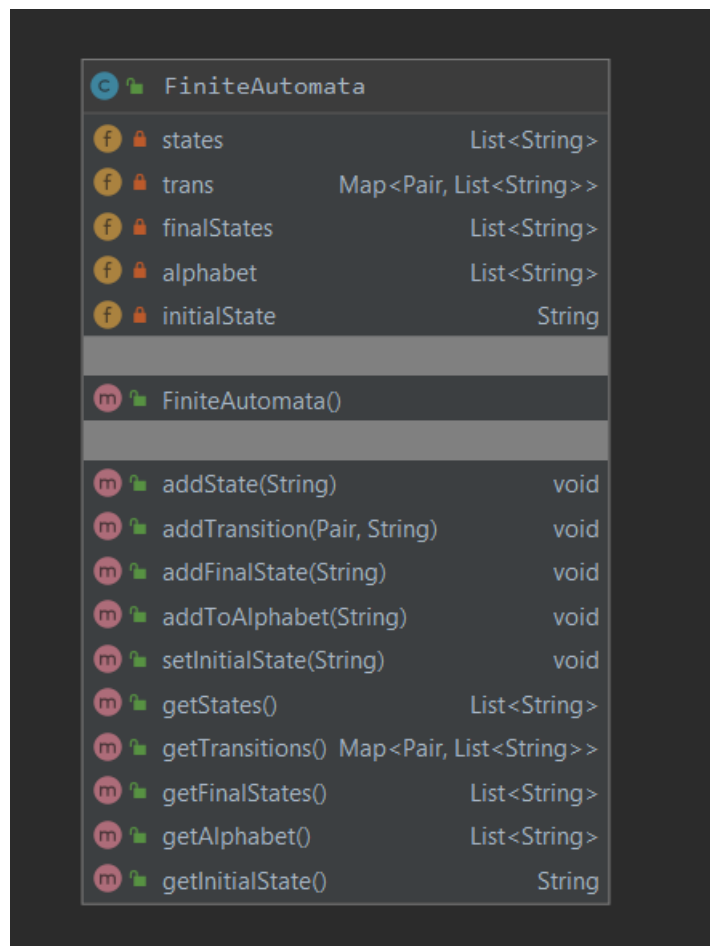
- States: list of states
- Final states: list of the final states
- Transitions: map with keys -> (startState, symbol) and values list of end states
- Alphabet: list of the alphabet
- Initial state: initial state

CLASS DIAGRAM

DFA CHECK

- The dfa check is being made by parsing through the map of transitions and checking if the value(list of strings which represents the states in which the start state with the symbol can transition into) has a greater size than 1, and in that case it returns false as it is not dfa
- otherwise it returns true

ACCEPTED SEQUENCE

- the check is being made by starting from the intial state and following the transition flow as long as the fa is a dfa and the flow matches the given transition
- if we get to a symbol that is not in the given sequence than we return false
- if the check is complete there is one last step to make, which is check if the last state that we reached is a final state

EXAMPLES:

1. fa.in

```
Q = { A, B, C }
E = { 0, 1 }
q0 = A
F = { C }
S = {(A;1)->A , (A;0)->B , (B;1)->A , (B;0)->C , (B;1)->C, (C;0)->C , (C;1)->C}
```

- States: A, B, C
- Alphabet: 0, 1
- Initial state: A
- Final state: C

```
Transitions: (A;1)->A , (A;0)->B , (B;1)->A , (B;0)->C , (B;1)->C, (C;0)->C ,
(C;1)->C
```

- DFA check:

```
Check DFA - failed
```

2. fa.in

```
Q = { A, B, C }
E = { 0, 1 }
q0 = A
F = { C }
S = {(A;1)->A , (A;0)->B , (B;1)->A , (B;0)->C , (C;0)->C , (C;1)->C}
```

- States: A, B, C
- Alphabet: 0, 1
- Initial state: A
- Final state: C
- Transitions:

```
S = {(A;1)->A , (A;0)->B , (B;1)->A , (B;0)->C , (C;0)->C , (C;1)->C}
```

- DFA check:

```
Check DFA - passed
```

3. fa.in

```
Q = { A, B, C }
E = { 0, 1 }
q0 = A
F = { C }
S = {(A;1)->A , (A;0)->B , (B;1)->A , (B;0)->C , (C;0)->C , (C;1)->C}
```

Sequence check:

001 - `Sequence accepted`

00101 - `Sequence accepted`

10 - `Sequence is NOT accepted`

4. fa.in

```
Q = { A, B, C }
E = { 0, 1 }
q0 = A
F = { C }
S = {(A;1)->A , (B;1)->A , (B;0)->C, (C;0)->C , (C;1)->C}
```

Sequence check:

101 - `Sequence is NOT accepted`

1 - `Sequence is NOT accepted`