

```
1  #include "Vigenere.h"
2
3  void Vigenere::initializeTable()
4  {
5      for (char row = 0; row < CHARACTERS; row++)
6      {
7          char lChar = 'B' + row;
8          for (char column = 0; column < CHARACTERS; column++)
9          {
10             if (lChar > 'Z')
11                 lChar = 'A';
12             fMappingTable[row][column] = lChar++;
13         }
14     }
15 }
16
17 Vigenere::Vigenere(const std::string& aKeyword): fKeyword(aKeyword),
18     fKeywordProvider(KeyProvider(aKeyword))
19 {
20     initializeTable();
21 }
22
23 std::string Vigenere::getCurrentKeyword()
24 {
25     std::string current_keyword;
26
27     for (size_t i = 0; i < fKeyword.length(); i++)
28     {
29         current_keyword += *fKeywordProvider;
30         fKeywordProvider << *fKeywordProvider;
31     }
32     return current_keyword;
33 }
34
35 void Vigenere::reset()
36 {
37     fKeywordProvider.initialize(fKeyword);
38 }
39
40 char Vigenere::encode(char aCharacter)
41 {
42     if (isalpha(aCharacter))
43     {
44         bool isLower = std::islower(aCharacter);
45         char encoded = fMappingTable[*fKeywordProvider - 'A'][std::toupper
46             (aCharacter) - 'A'];
47
48         fKeywordProvider << aCharacter;
49         if (isLower)
```

```
48     {
49         return static_cast<char>(std::tolower(encoded));
50     }
51     return encoded;
52 }
53 return aCharacter;
54 }
55
56 char Vigenere::decode(char aCharacter)
57 {
58     if (isalpha((aCharacter)))
59     {
60         bool isLower = std::islower(aCharacter);
61         char encoded = static_cast<char>(toupper(aCharacter));
62         char decoded = 0;
63
64         for (char column = 0; column < CHARACTERS; column++)
65         {
66             if (fMappingTable[*fKeywordProvider - 'A'][column] == encoded)
67             {
68                 decoded = static_cast<char>(column + 'A');
69                 break;
70             }
71         }
72
73         fKeywordProvider << decoded;
74         if (isLower)
75         {
76             return static_cast<char>(std::tolower(decoded));
77         }
78         return decoded;
79     }
80     return aCharacter;
81 }
```