



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 10
по курсу «Языки и методы программирования»
«Реализация итераторов на языке C++»

Студент группы ИУ9-21Б Горбунов А. Д.

Преподаватель Посевин Д. П.

Москва 2023

1 Задание

Строка, составленная из латинских букв, с константным однонаправленным итератором по максимальным «правильным» подстрокам.

«Правильная» подстрока должна содержать либо исключительно гласные, либо исключительно согласные буквы.

2 Результаты

Исходный код программы представлен в листинге 1, 2, 3

Листинг 1 — main.cpp

```
1 #include "Substrings.h"
2 using namespace std;
3
4 int main() {
5     Substrings s1("feegeeeozd UAIEOoeegreeju
6     uqwertyuiopasdfghjklzxcvbnm");
7     for (Substrings::Iterator it = s1.begin(); it != s1.end(); ++it) {
8         cout << *it << endl;
9     }
10    cout<<endl;
11    Substrings s2("ee eeeo UAIEOooo ee uu e uio a");
12    for (Substrings::Iterator it = s2.begin(); it != s2.end(); ++it) {
13        cout << *it << endl;
14    }
15    cout<<endl;
16    Substrings s3("dOuBIE fReE ");
17    for (Substrings::Iterator it = s3.begin(); it != s3.end(); ++it) {
18        cout << *it << endl;
19    }
20    return 0;
}
```

Листинг 2 — Substrings.h

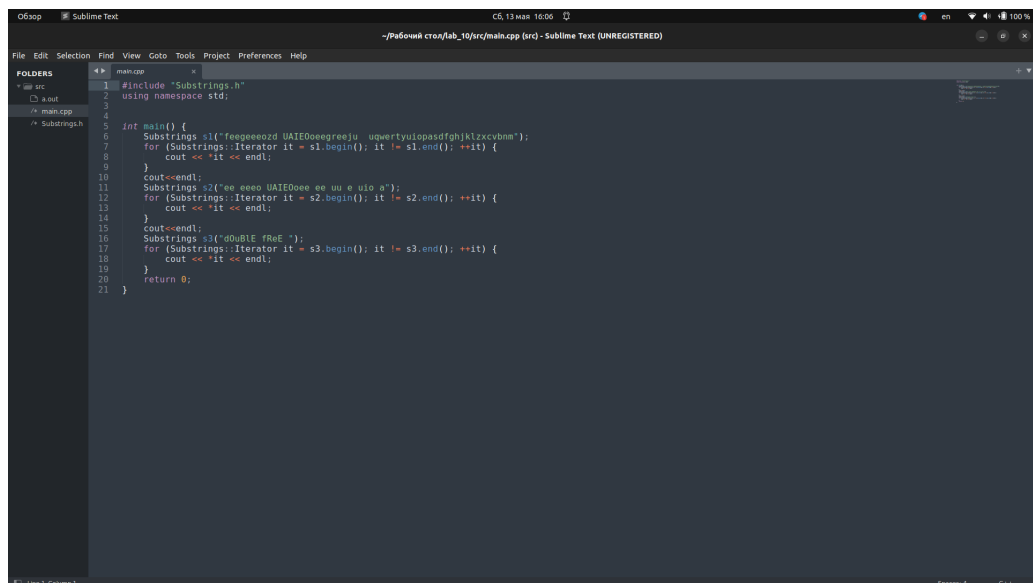
```

1 #include <iostream>
2 #include <string>
3 #include <iterator>
4 using namespace std;
5 class Substrings {
6 public:
7     Substrings(const string& str) : str_(str) {}
8     class Iterator : public iterator<forward_iterator_tag, string> {
9     public:
10         unsigned int start_pos = 0;
11         string substring;
12
13         Iterator(const string& str, unsigned int pos) : str_(str),
14         start_pos(pos) {
15             while (start_pos < str_.length() && !isVowel(str_[start_pos
16             ])) {
17                 start_pos++;
18             }
19             if (start_pos == str_.length()) {
20                 start_pos = 0;
21                 substring = "";
22             }
23             else {
24                 unsigned int end_pos = start_pos + 1;
25                 while (end_pos < str_.length() && isVowel(str_[end_pos])
26                 ) {
27                     end_pos++;
28                 }
29                 substring = str_.substr(start_pos, end_pos - start_pos);
30                 start_pos = end_pos;
31             }
32         }
33
34         bool operator==(const Iterator& other) const {
35             return start_pos == other.start_pos && substring == other.
36             substring;
37         }
38
39         bool operator!=(const Iterator& other) const {
40             return !(*this == other);
41         }
42
43         Iterator& operator++() {
44             while (start_pos < str_.length() && !isVowel(str_[start_pos
45             ])) {
46                 start_pos++;
47             }
48             if (start_pos == str_.length()) {
49                 start_pos = 0;
50                 substring = "";
51             }
52             else {
53                 unsigned int end_pos = start_pos + 1;
54                 while (end_pos < str_.length() && isVowel(str_[end_pos])
55                 ) {
56                     end_pos++;
57                 }
58                 substring = str_.substr(start_pos, end_pos - start_pos);
59                 start_pos = end_pos;
60             }
61             return *this;
62         }
63     }
64 }
```

Листинг 3 — Substrings.h(продолжение)

```
1      Iterator operator++(int) {
2          Iterator temp = *this;
3          ++(*this);
4          return temp;
5      }
6      const string& operator*() const {
7          return substring;
8      }
9  private:
10     const string str_;
11
12     bool isVowel(char c) const {
13         return c == 'a' || c == 'e' || c == 'i' || c == 'o' || c ==
14         'u' || c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U';
15     };
16     Iterator begin() const {
17         return Iterator(str_, 0);
18     }
19     Iterator end() const {
20         return Iterator(str_, str_.length());
21     }
22 private:
23     const string str_;
24 };
```

Результат запуска представлен на рисунке 1, 2, 3



```
1  #include "Substrings.h"
2  using namespace std;
3
4  int main() {
5      Substrings s1("feegewozd UAIEDoeegeeju qwertyuiopasdfghiklxcvbnm");
6      for (Substrings::Iterator it = s1.begin(); it != s1.end(); ++it) {
7          cout << *it << endl;
8      }
9      cout << endl;
10     Substrings s2("ee eeeo UAIEDoe ee uu e uio a");
11     for (Substrings::Iterator it = s2.begin(); it != s2.end(); ++it) {
12         cout << *it << endl;
13     }
14     cout << endl;
15     Substrings s3("dOUBLE rReE ");
16     for (Substrings::Iterator it = s3.begin(); it != s3.end(); ++it) {
17         cout << *it << endl;
18     }
19     return 0;
20 }
```

Рис. 1 — Реализация main.cpp

