



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 ИУ9-22Б Тараканов В. Д.

Преподаватель Посевин Д. П.

Москва 2024

1 Задание

Последовательность n-мерных векторов с константным двунаправленным итератором по скалярным произведениям соседних векторов. Обращение к элементам последовательности должно осуществляться с помощью перегруженной операции «[]».

2 Результаты

Листинг 1 — Вспомогательный класс Vector

```
1      #ifndef VECTOR_H
2      #define VECTOR_H
3
4      #include <vector>
5
6      class Vector {
7      public:
8          Vector(const std::vector<double>& components);
9
10         int size() const;
11         double operator[](int index) const;
12
13     private:
14         std::vector<double> components;
15     };
16
17     #endif // VECTOR_H
18
```

Листинг 2 — Реализация методов класса Vector

```
1      #include "Vector.h"
2
3      Vector::Vector(const std::vector<double>& components) : components
(components) {}
4
5      int Vector::size() const {
6          return components.size();
7      }
8
9      double Vector::operator[](int index) const {
10
11          return components[index];
12      }
13
```

Листинг 3 — Класс VectorSequence

```
1  #ifndef VECTORSEQUENCE_H
2  #define VECTORSEQUENCE_H
3
4  #include <vector>
5  #include "Vector.h"
6  #include <iterator>
7
8
9  class VectorSequence {
10 public:
11     void addVector(const Vector& vector);
12
13     class ConstIterator : public std::iterator<std::
bidirectional_iterator_tag, double> {
14     public:
15         ConstIterator(const VectorSequence& sequence, int index);
16
17         bool operator==(const ConstIterator& other) const;
18         bool operator!=(const ConstIterator& other) const;
19
20         double operator*() const;
21
22         ConstIterator& operator++();
23         ConstIterator operator++(int);
24
25         ConstIterator& operator--();
26         ConstIterator operator--(int);
27
28     private:
29         const VectorSequence& sequence;
30         int index;
31
32         double dotProduct(const Vector& v1, const Vector& v2) const;
33     };
34
35     ConstIterator begin() const;
36     ConstIterator end() const;
37
38     private:
39     std::vector<Vector> vectors;
40 };
41
42 #endif // VECTORSEQUENCE_H
43
```

Листинг 4 — Реализация методов класса VectorSequence

```
1  #include "VectorSequence.h"
2
3  void VectorSequence::addVector(const Vector& vector) {
4      vectors.push_back(vector);
5  }
6
7  VectorSequence::ConstIterator::ConstIterator(const VectorSequence&
8  sequence, int index)
9      : sequence(sequence), index(index) {}
10
11 bool VectorSequence::ConstIterator::operator==(const ConstIterator
12 & other) const {
13     return &sequence == &other.sequence && index == other.index;
14 }
15
16 bool VectorSequence::ConstIterator::operator!=(const ConstIterator
17 & other) const {
18     return !(*this == other);
19 }
20
21 double VectorSequence::ConstIterator::operator*() const {
22     return dotProduct(sequence.vectors[index], sequence.vectors[
23 index + 1]);
24 }
25
26 VectorSequence::ConstIterator& VectorSequence::ConstIterator::
27 operator++() {
28     ++index;
29     return *this;
30 }
31
32 VectorSequence::ConstIterator VectorSequence::ConstIterator::
33 operator++(int) {
34     ConstIterator temp = *this;
35     ++(*this);
36     return temp;
37 }
38
39 VectorSequence::ConstIterator& VectorSequence::ConstIterator::
40 operator--() {
41     --index;
42     return *this;
43 }
44
45 VectorSequence::ConstIterator VectorSequence::ConstIterator::
46 operator--(int) {
47     ConstIterator temp = *this;
48     --(*this);
49     return temp;
50 }
51
52 }
```

Листинг 5 — Реализация методов класса VectorSequence(продолжение)

```
1      double VectorSequence::ConstIterator::dotProduct(const Vector& v1,  
2      const Vector& v2) const {  
3          double result = 0.0;  
4          for (int i = 0; i < v1.size(); ++i) {  
5              result += v1[i] * v2[i];  
6          }  
7          return result;  
8      }  
9  
10     VectorSequence::ConstIterator VectorSequence::begin() const {  
11         return ConstIterator(*this, 0);  
12     }  
13  
14     VectorSequence::ConstIterator VectorSequence::end() const {  
15         return ConstIterator(*this, vectors.size() - 1);  
16     }  
17
```

Листинг 6 — Функция main, проверяющая работу класса VectorSequence

```
1      #include <iostream>  
2      #include "Vector.h"  
3      #include "VectorSequence.h"  
4  
5      int main() {  
6          VectorSequence sequence;  
7          sequence.addVector(Vector({1.0, 1.0, 1.0}));  
8          sequence.addVector(Vector({1.0, 1.0, 1.0}));  
9          sequence.addVector(Vector({7.0, 8.0, 9.0}));  
10  
11         for (auto it = sequence.begin(); it != sequence.end(); ++it) {  
12             std::cout << *it << " ";  
13         }  
14  
15  
16         return 0;  
17     }  
18  
19
```

Результат запуска представлен на рисунке 1.



Рис. 1 — Результат