



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 № 4
по курсу «Языки и методы программирования»
«Реализация итераторов в языке Java»

Студент группы ИУ9-22Б Тараканов В. Д.

Преподаватель Посевин Д. П.

Москва 2024

1 Задание

Последовательность векторов в n -мерном пространстве и итератором по максимальным по длине непрерывным подпоследовательностям, составленных из взаимно ортогональных векторов. (Размерность n пространства задаётся параметром конструктора.)

2 Результаты

Листинг 1 — Вспомогательный класс Vector

```
1  import java.util.Arrays;
2
3  class Vector {
4      private int[] coordinates;
5      private int dimension;
6      public Vector(int dimension, int... coordinates) {
7          this.dimension = dimension;
8          this.coordinates = coordinates;
9      }
10
11     public boolean isOrthogonalTo(Vector other) {
12         if (this.coordinates.length != other.coordinates.length) {
13             return false;
14         }
15
16         int dotProduct = 0;
17         for (int i = 0; i < coordinates.length; i++) {
18             dotProduct += other.coordinates[i] * this.coordinates[i];
19         }
20         return dotProduct == 0;
21     }
22
23     public String toString() {
24         return "Vector{ coordinates = " + Arrays.toString(coordinates)
25 + '}'';
26     }
27 }
28
```

Листинг 2 — Класс `OrthogonalVectorSequence`, в котором реализована программа по заданию

```
1      import java.util.ArrayList;
2      import java.util.Iterator;
3      import java.util.List;
4
5      public class OrthogonalVectorSequence {
6
7          private List<Vector> vectors;
8
9          public OrthogonalVectorSequence(int dimension) {
10
11              vectors = new ArrayList<>();
12          }
13
14          public void addVector(Vector vector) {
15              vectors.add(vector);
16          }
17
18          public Iterator<List<Vector>> getOrthogonalSubsequencesIterator
19          () {
20              return new OrthogonalSubsequencesIterator();
21          }
22
23          private class OrthogonalSubsequencesIterator implements Iterator
24          <List<Vector>> {
25              private int currentIndex = 0;
26
27              public boolean hasNext() {
28                  return currentIndex < vectors.size();
29              }
30
31              public List<Vector> next() {
32                  if (!hasNext()) {
33                      return null;
34                  }
35
36                  List<Vector> subsequence = new ArrayList<>();
37                  subsequence.add(vectors.get(currentIndex));
38                  int nextIndex = currentIndex + 1;
39
40                  while (nextIndex < vectors.size() && vectors.get(nextIndex).
41                  isOrthogonalTo(subsequence.get(subsequence.size() - 1))) {
42                      subsequence.add(vectors.get(nextIndex));
43                      nextIndex++;
44                  }
45
46                  currentIndex = nextIndex;
47                  return subsequence;
48              }
49          }
50      }
```

Листинг 3 — Класс Main, в котором реализована проверка работы класса OrthogonalVectorSequence

```
1  import java.util.Iterator;
2  import java.util.List;
3
4  public class Main {
5      public static void main(String[] args) {
6          OrthogonalVectorSequence sequence = new OrthogonalVectorSequence
7              (3);
8          sequence.addVector(new Vector(3,1, 0, 0));
9          sequence.addVector(new Vector(3,0, 1, 0));
10         sequence.addVector(new Vector(3,1, 1, 0));
11         sequence.addVector(new Vector(3,-1, 1, 0));
12         sequence.addVector(new Vector(3,1, 10, 0));
13         sequence.addVector(new Vector(3,8, 344, 0));
14
15         Iterator<List<Vector>> iterator = sequence.
16             getOrthogonalSubsequencesIterator();
17         while (iterator.hasNext()) {
18             List<Vector> subsequence = iterator.next();
19             if (subsequence.size()>1) {
20                 System.out.println(subsequence);
21             }
22         }
23     }
```

Результат запуска представлен на рисунке 1.

```
[Vector{ coordinates = [1, 0, 0]}, Vector{ coordinates = [0, 1, 0]}]
[Vector{ coordinates = [1, 1, 0]}, Vector{ coordinates = [-1, 1, 0]}]
```

Рис. 1 — Результат