



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 Федуков А. А.

Преподаватель Посевин Д. П.

28 марта 2024 г.

Цель работы

Изучение обобщённых итераторов и экземплярных вложенных классов языка Java.

Задание

Во время выполнения лабораторной работы требуется разработать на языке Java один из классов, перечисленных в таблицах 1 – 7. Класс должен реализовывать интерфейс `Iterable`. Объект разрабатываемого класса должен быть изменяемым, то есть в нём надо так или иначе предусмотреть возможность изменения внутреннего состояния. В методе `main` вспомогательного класса `Test` нужно продемонстрировать работоспособность разработанного класса.

Задание 1

Изменяемая строка с итератором по индексам первых букв вхождений заданной подстроки `w` (`w` задаётся в конструкторе).

Реализация

Я описал класс `SuffixList`, исходя из условия задания, в файле [SuffixList.java](#). Сгенерировал его экземпляры и проверил работоспособность уже в [Test.java](#).

Код

Листинг 1 — Файл SuffixList.java

```
1 package task1;
2
3 import java.util.Iterator;
4 import java.util.Objects;
5
6 public class SuffixList implements Iterable<Integer> {
7     private StringBuilder s;
8     private String w;
9
10    public void setW(String s){
11        this.w = s;
12    }
13    public SuffixList ( StringBuilder s , String w) {
14        this.s = s;
15        this.w = w;
16    }
17    public Iterator<Integer> iterator() { return new SuffixIterator(); }
18    private class SuffixIterator implements Iterator<Integer> {
19        private int pos;
20
21        private Integer GetNext(){
22            int tpos = pos + 1;
23            while (tpos <= s.length() - w.length()){
24                if (Objects.equals(s.substring(tpos, tpos + w.length()),
25w)){
26                    return tpos;
27                }
28                tpos += 1;
29            }
30            return -1;
31        }
32        public SuffixIterator () { pos = 0 ; }
33        public boolean hasNext () { return GetNext() != -1; }
34        public Integer next () {
35            int tpos = GetNext();
36            pos = tpos + 1;
37            return tpos;
38        }
39    }
```

Листинг 2 — Файл Test.java

```
1 package task1;
2
3 public class Test {
4     public static void main(String[] args) {
5         StringBuilder b = new StringBuilder("
6         kmqwertyjkdndsnjvdjwekmcvklvcwekmcvkcvcnvnjcnjwe");
7         SuffixList suff = new SuffixList(b, "we");
8         for(Integer s : suff) System.out.println(s + " " +b.substring(s,
9         b.length()));
10
11         System.out.println("AFTER Changes");
12         b.insert(1,"x");
13         suff.setW("km");
14
15         for(Integer s : suff) System.out.println(s + " " +b.substring(s,
16         b.length()));
17     }
18 }
```

Вывод программы

Программа создала строку, вывела индексы ее первых букв вхождений заданной подстроки w. Причем, после изменения меньшей строки эта функциональность также сохранилась.

Листинг 3 — Вывод программы

```
1 3 wertyjkdndsnjvdjwekmcvklvcwekmcvkcvcnvnjcnjwe
2 19 wekmcvklvcwekmcvkcvcnvnjcnjwe
3 29 wekmcvkcvcnvnjcnjwe
4 45 we
5 AFTER Changes
6 22 kmcvklvcwekmcvkcvcnvnjcnjwe
7 32 kmvcvkcvcnvnjcnjwe
8
9 Process finished with exit code 0
```

Задание 2

Последовательность строк с итератором по максимальным суффиксам, совпадающим с префиксом следующей строки последовательности.

Реализация

Я создал класс [StringList](#), принимающий список строк.

Сгенерировал его экземпляры и проверил работоспособность итератора уже в [Test.java](#)

Код

Листинг 4: Файл MyInt.java

```
1 package task2;
2
3 import java.util.Iterator;
4
5 public class StringList implements Iterable<String> {
6     private String[] strings;
7
8     public void setStrings(String[] s){
9         this.strings = s;
10    }
11    public StringList(String[] s){
12        this.strings = s;
13    }
14    private class StringListIterator implements Iterator<String> {
15        private int stringIndex;
16        public StringListIterator () {stringIndex = 0;}
17
18
19        public boolean hasNext () {
20            for (int j = stringIndex; j < strings.length - 1; j++) {
21                for (int i = 0; i < strings[j].length(); i++) {
22                    String sub = strings[j].substring(i);
23                    if (strings[j+1].startsWith(sub)){
24                        return true;
25                    }
26                }
27            }
28            return false;
```

```

29     }
30     public String next () {
31         for (; stringIndex < strings.length - 1; stringIndex++) {
32             for (int i = 0; i < strings[stringIndex].length(); i++)
33             {
34                 String sub = strings[stringIndex].substring(i);
35                 if (strings[stringIndex+1].startsWith(sub)){
36                     stringIndex += 1;
37                     return sub;
38                 }
39             }
40             return "";
41         }
42     }
43     @Override
44     public Iterator<String> iterator () {
45         return new StringListIterator();
46     }
47 }

```

Листинг 5 — Файл Test.java

```

1 package task2;
2
3 public class Test {
4     public static void main(String[] args ) {
5         String[] s = new String[5];
6         s[0] = "abhcsa1";
7         s[1] = "alaaaaa";
8         s[2] = "bbbbbaaa2";
9         s[3] = "aaa2aaadsf";
10        s[4] = "sfaaaaacc";
11
12        StringList L = new StringList(s);
13        for (String suff: L) System.out.println(suff);
14
15        System.out.println("AFTER Changes");
16
17        s[0] = "zxc";
18        for (String suff: L) System.out.println(suff);
19    }
20 }
21 }

```

Вывод программы

Программа выдала пересечения рядом стоящих строк

Листинг 6 — Вывод программы

```
1 a1
2 aaa2
3 sf
4 AFTER Changes
5 aaa2
6 sf
```

Вывод

На этот раз я научился реализовывать собственные итераторы классов для более удобного взаимодействия с их экземплярами.