



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. Вариант 5»**

Студент группы ИУ9-22Б Жук А. Д.

Преподаватель Посевин Д. П.

*Москва 2023*

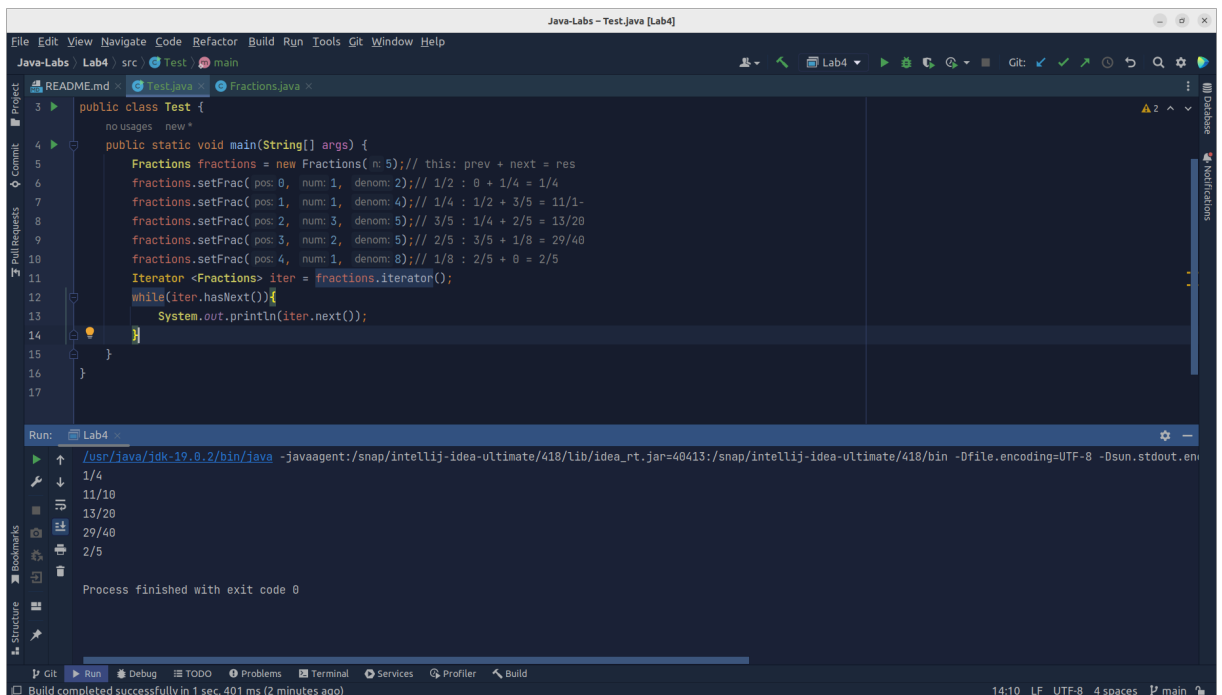
# 1 Задание

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.

5. Последовательность нормализованных дробей с итератором по суммам соседних дробей.

## 2 Результаты

Создадим массив дробей (`Fractions`), добавим туда 5 дробей. Проверим работу программы.



```
File Edit View Navigate Code Refactor Build Run Tools Git Window Help
Java-Labs Lab4 src Test main
README.md Test.java Fractions.java
3 public class Test {
4     no usages new *
5     public static void main(String[] args) {
6         Fractions fractions = new Fractions(5); // this: prev + next = res
7         fractions.setFrac(pos: 0, num: 1, denom: 2); // 1/2 : 0 + 1/4 = 1/4
8         fractions.setFrac(pos: 1, num: 1, denom: 4); // 1/4 : 1/2 + 3/5 = 11/10
9         fractions.setFrac(pos: 2, num: 3, denom: 5); // 3/5 : 1/4 + 2/5 = 13/20
10        fractions.setFrac(pos: 3, num: 2, denom: 5); // 2/5 : 3/5 + 1/8 = 29/40
11        fractions.setFrac(pos: 4, num: 1, denom: 8); // 1/8 : 2/5 + 0 = 2/5
12        Iterator<Fractions> iter = fractions.iterator();
13        while(iter.hasNext()){
14            System.out.println(iter.next());
15        }
16    }
17 }
Run: Lab4
/usr/java/jdk-19.0.2/bin/java -javaagent:/snap/intellij-idea-ultimate/418/lib/idea_rt.jar=40413:/snap/intellij-idea-ultimate/418/bin -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8
1/4
11/10
13/20
29/40
2/5
Process finished with exit code 0
Build completed successfully in 1 sec, 401 ms (2 minutes ago) 14:10 LF UTF-8 4 spaces main
```

Рис. 1 — Скриншот работы программы

Далее проверим работоспособность, изменив некоторые дроби.

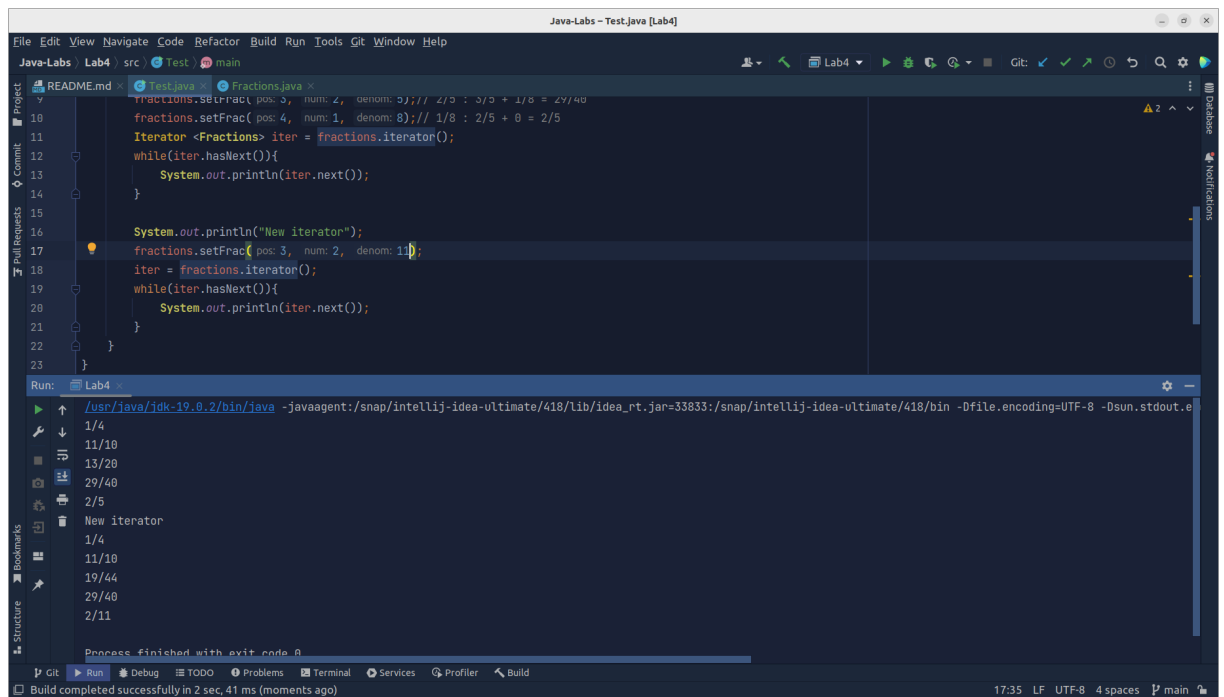


Рис. 2 — Скриншот перерасчёта

Также реализовано упрощение дробей (внимание на первый вывод)

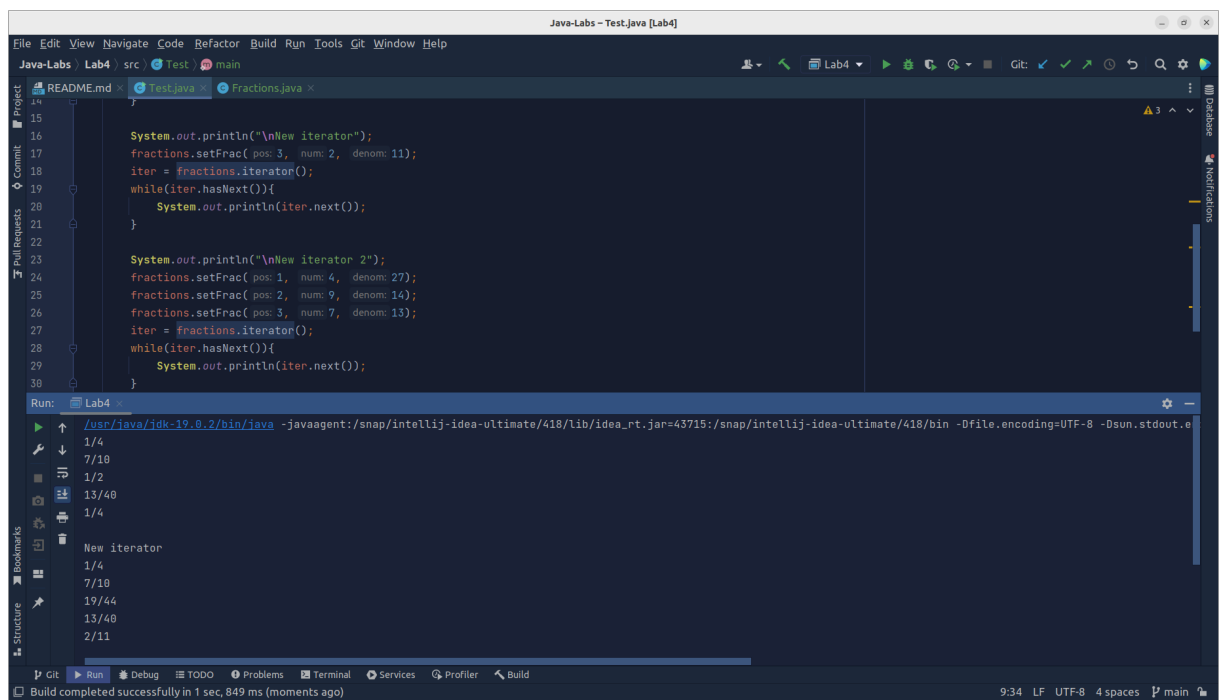


Рис. 3 — Скриншот перерасчёта с упрощением

## Листинг 1 — Код класса Fractions

```

1 import java.util.Iterator;
2 import java.util.NoSuchElementException;
3
4 public class Fractions implements Iterable {
5     public Fraction[] fractions;
6
7     Fractions(int n) {
8         this.fractions = new Fraction[n];
9         for (int i = 0; i < n; i++) {
10             this.fractions[i] = new Fraction(0, 1);
11         }
12     }
13     public void setFrac(int pos, int num, int denom) {
14         this.fractions[pos].num = num;
15         this.fractions[pos].denom = denom;
16     }
17     public Fraction getFrac(int pos) {
18         return fractions[pos];
19     }
20     public Iterator iterator() {
21         return new FractionsIterator();
22     }
23     private class Fraction {
24         int num;
25         int denom;
26         Fraction(int num, int denom) {
27             this.num = num;
28             this.denom = denom;
29         }
30         public String toString() {
31             StringBuilder result = new StringBuilder();
32             result.append(this.num + "/" + this.denom);
33             return result.toString();
34         }
35         private int gcdByEuclidsAlgorithm(int n1, int n2) {
36             if (n2 == 0) {
37                 return n1;
38             }
39             return gcdByEuclidsAlgorithm(n2, n1 % n2);
40         }
41         public void simplify() {
42             int gcd = gcdByEuclidsAlgorithm(this.num, this.denom);
43             this.num /= gcd;
44             this.denom /= gcd;
45         }
46     }
47     private class FractionsIterator implements Iterator {
48         private int pos = 0;
49         public boolean hasNext() {
50             return pos < fractions.length;
51         }
52         public Fraction next() {
53             Fraction prevF, nextF;
54             if (pos > 0) {
55                 prevF = fractions[pos - 1];
56             } else {prevF = new Fraction(0, 1);}
57             if (pos < fractions.length - 1) {
58                 nextF = fractions[pos + 1];
59             } else {nextF = new Fraction(0, 1);}
60             int resD = nextF.denom * prevF.denom;
61             int resN = (prevF.num * (resD / prevF.denom)) + (nextF.num *
62 (resD / nextF.denom));
63             pos++;
64             Fraction res = new Fraction(resN, resD);
65             res.simplify();
66             return res;
67         }
68     }
69 }

```

## Листинг 2 — Код класса Test

```
1 import java.util.Iterator;
2
3 public class Test {
4     public static void main(String[] args) {
5         Fractions fractions = new Fractions(5); // this: prev + next =
        res
6         fractions.setFrac(0, 1, 2); // 1/2 : 0 + 1/4 = 1/4
7         fractions.setFrac(1, 1, 4); // 1/4 : 1/2 + 3/5 = 11/10
8         fractions.setFrac(2, 1, 5); // 3/5 : 1/4 + 2/5 = 13/20
9         fractions.setFrac(3, 1, 4); // 2/5 : 3/5 + 1/8 = 29/40
10        fractions.setFrac(4, 1, 8); // 1/8 : 2/5 + 0 = 2/5
11        Iterator <Fractions> iter = fractions.iterator();
12        while(iter.hasNext()){
13            System.out.println(iter.next());
14        }
15
16        System.out.println("\nNew iterator");
17        fractions.setFrac(3, 2, 11);
18        iter = fractions.iterator();
19        while(iter.hasNext()){
20            System.out.println(iter.next());
21        }
22
23        System.out.println("\nNew iterator 2");
24        fractions.setFrac(1, 4, 27);
25        fractions.setFrac(2, 9, 14);
26        fractions.setFrac(3, 7, 13);
27        iter = fractions.iterator();
28        while(iter.hasNext()){
29            System.out.println(iter.next());
30        }
31    }
32 }
33 }
```