

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)

Факультет: Информатика и системы управления

Кафедра: Теоретическая информатика и компьютерные технологии

Лабораторная работа №4-1
по курсу: «Языки и методы программирования»
«Реализация итераторов в языке Java»

Выполнил:
Студент группы ИУ9-21Б

Проверил:
Посевин Д. П.

Москва, 2024

1. Цель

Изучение обобщённых итераторов и экземплярных вложенных классов языка Java.

2. Персональный вариант

Множество дробей с итератором по всем парам дробей, произведение которых равно 1.

3. Решение

3.1. Код

```
import java.util.Random;

public class Test {
    public static void main(String[] args) {
        Random r = new Random();
        FractionsSet s = new FractionsSet();

        for (int i = 0; i < 10; ++i) {
            int a = r.nextInt(-10, 10), b = r.nextInt(-10, 10);
            s.addFraction(new Fraction(a, b));
        }
        s.print();
    }
}
```

Код 1: Test.java

```
public class Pair<T, U> {

    private T first;
    private U second;

    public Pair(T first, U second) {
        this.first = first;
        this.second = second;
    }

    public T getFirst() {
        return first;
    }
}
```

```

    public U getSecond() {
        return second;
    }
}

```

Код 2: Pair.java

```

public class Fraction {
    Integer a, b;

    public Fraction(Integer a, Integer b) {
        this.a = a;
        this.b = b;
        this.simplify();
    }

    public Pair<Integer, Integer> toPair() { return new
    ↪ Pair<>(a, b); }

    public void print() { System.out.printf("%d/%d ", this.a,
    ↪ this.b); }

    private Integer GCD(Integer a, Integer b) {
        while (a != 0 && b != 0) {
            if (a > b) a %= b;
            else b %= a;
        }
        return a + b;
    }

    private Integer ABS(Integer a) {
        return (a < 0 ? -a : a);
    }

    private void simplify() {
        int g = GCD(ABS(a), ABS(b));

        if (b < 0) {
            a /= -1;
            b /= -1;
        }

        if (g != 0) {
            b /= g;

```

```

        a /= g;
    }
}

public Fraction multiply(Fraction other) {
    int A = this.a * other.a;
    int B = this.b * other.b;
    return new Fraction(A, B);
}
}

```

Код 3: Fraction.java

```

import java.util.ArrayList;
import java.util.Iterator;

public class FractionsSet implements Iterable<Pair<Fraction,
↪ Fraction>> {
    private ArrayList<Fraction> fractions;

    public FractionsSet() {
        this.fractions = new ArrayList<>();
    }

    public void addFraction(Fraction f) {
        this.fractions.add(f);
    }

    public void print() {
        System.out.println("Current fractions:");
        for (Fraction f : fractions) {
            f.print();
            System.out.println();
        }

        System.out.println("Pairs with * of 1:");
        int count = 0;
        for (Pair<Fraction, Fraction> f : this) {
            System.out.printf("%d: ", ++count);
            f.getFirst().print();
            f.getSecond().print();
            System.out.println();
        }
    }
}

```

```

public Iterator<Pair<Fraction, Fraction>> iterator() {
    ↪ return new SetIterator(); }

private class SetIterator implements Iterator<Pair<Fraction,
    ↪ Fraction>> {
    private int pos1, pos2;
    private boolean hasPairs, recalculate;

    public SetIterator() {
        this.hasPairs = false;
        this.recalculate = true;
        this.pos1 = this.pos2 = 0;
    }

    public void calculateNextPair() {
        if (!recalculate)
            return;

        recalculate = false;

        if (pos2 >= pos1) {
            pos2 = 0;
            pos1++;
        }

        hasPairs = false;
        for (; pos1 < fractions.size(); ++pos1) {
            Fraction f = fractions.get(pos1);
            for (; pos2 < pos1; ++pos2) {
                Fraction g = fractions.get(pos2);
                Pair<Integer, Integer> r =
                    ↪ g.multiply(f).toPair();
                if (r.getFirst() == 1 && r.getSecond() == 1)
                    ↪ {
                        hasPairs = true;
                        break;
                    }
            }

            if (hasPairs)
                break;

            pos2 = 0;
        }
    }
}

```

```

    }

    public boolean hasNext() {
        calculateNextPair();
        return hasPairs;
    }

    public Pair<Fraction, Fraction> next() {
        calculateNextPair();

        if (hasPairs) {
            recalculate = true;
            return new Pair<>(fractions.get(pos1),
                ↪ fractions.get(pos2++));
        }

        return new Pair<>(new Fraction(0, 0), new
            ↪ Fraction(0, 0));
    }
}

```

Код 4: FractionsSet.java

3.2. Скриншоты

Current fractions:

$5/4$

$6/5$

$2/3$

$8/1$

$-8/5$

$2/3$

$-2/3$

$-4/5$

$-5/8$

$-5/3$

Pairs with * of 1:

1: $-5/8$ $-8/5$

Process finished with exit code 0

Рис. 1: Пример работы