Общее задание

- Составить на языке Java описания классов для указанных объектов.
- Разработать консольную программу, иллюстрирующую использование объектов заданных классов.

Индивидуальное задание

Рациональная дробь вида P(x)/Q(x) на основе двунаправленного связанного списка (сложение, умножение).

Описание работы программы

Перед пользователем возникает окно консольного приложения в котором есть список выполняемых действий. По номеру действия его можно вызвать. Первое действие обеспечивает ввод двух рациональных дробей, второе их печатает, третье - печатаем сумму введенных дробей, четвертое - их произведение. Можно неограниченное число раз вводить разные дроби и выполнять над ними действия.

Алгоритмы выполнения основных операций на псевдокоде

Произведение двух дробей.
Числ_новой_дроби;
Знам_новой_дроби;
*Числ_новой_дроби = Числ_дроби_A * Числ_дроби_B;*
*Знам_новой_дроби = Знам_дроби_A * Знам_дроби_B;*

Сумма двух дробей.
Числ_новой_дроби;
Знам_новой_дроби;
*Числ_новой_дроби = Числ_дроби_A * Знам_дроби_B + Числ_дроби_B * Знам_дроби_A;*
*Знам_новой_дроби = Знам_дроби_A * Знам_дроби_B;*

Так как происходят операции с многочленами, то приведем их алгоритмы на псевдокоде.

Сложение двух многочленов
AdditionPolynom(Polynom S)

```
p = first;
ПОКА(p != null)
    sum = p.value;
    p_s = S.first;
    ПОКА(p_s != null)
            ЕСЛИ(p.degree == p_s.degree)
                    sum += p_s.value;
                    p_s_2 = p_s.next;
                    S.deleteNode(p.s);
                    p_s = p_s_2;
            ИНАЧЕ
                    p_s = p_s.next;
    ЕСЛИ(sum != 0)
            result.AddNode(sum,p.degree);
    p = p.next;
p = S.first;
ПОКА(p != null)
    result.AddNode(p.value, p.degree);
    p = p.next;
ВЫВОД result;
```

Произведение двух многочленов

```
MultiplicationPolynom(Polynom S)
p = first;
ПОКА(p != null)
p_s = S.first;
ПОКА(p_s != null)
                    sum_value = p_s.value*p_value;
                    sum_degree = p.degree + p_s.degree;
                    result.AddNode(sum_value, sum_degree)
                    p_s = p_s.next;
p = p.next;
ВЫВОД result;
```

Тесты

№ Теста	**Входные данные**	**Выходные данные**

1	**Дробь A:** 1x^(1)-9x^(0) ---------- 1x^(2)-9x^(0) **Дробь B:** -3x^(0) ---------- 3x^(1)-1x^(2)	**Сложение:** 9x^(2)-1x^(3)-27x+27 ---------- 3x^(3)-1x^(4)-27x+9x^(2) **Произведение:** -3x+27 ---------- 3x^(3)-1x^(4)-27x^(1)+9x^(2)
2	**(Происходит приведение подобных слагаемых в числителе)** **Дробь A:** 3x^(4)-14x^(8) ---------- 7x+2x^(2)+3x^(3) **Дробь B:** 6x^(9) ---------- 3x^(2)-4x^(7)	**Сложение:** 9x^(6)+56x^(15)+18x^(12) ---------- 21x^(3)-28x^(8)+6x^(4)-8x^(9)+9x^(5)-12x^(10) **Произведение:** 18x^(13)-84x^(17) ---------- 21x^(3)-28x^(8)+6x^(4)-8x^(9)+9x^(5)-12x^(10)
3	**(Числитель обращается в нуль)** **Дробь A:** 1x^(5)+3x^(6)1x^(4) ---------- 1x^(4) **Дробь B:** -1x-3x^(2)1 ---------- 1	**Сложение:** 0 ---------- **Произведение:** -1x^(6)-6x^(7)-9x^(8)1x^(4) ---------- 1x^(4)
4	**(Знаменатель обращается в нуль)** **Дробь A:** 2x^(2)+3x^(2)	**Сложение:** **Знаменатель обратился в нуль =(**

	---------- 3x^(5)-3x **Дробь B:** 2x^(2) ---------- 4x^(3)-4x^(3)	---------- --------- **Произведение:** **Знаменатель обратился в нуль =(** ---------- ---------

Распечатки экранов при работе программы

Тест 4

```
1 - Ввести рациональные дроби
2 - Вывести рациональные дроби
3 - Вывести результат сложения дробей
4 - Вывести результат перемножения дробей
5 - Завершить программу
Выберите действие ---- 1
-+-+-+Введем числитель+-+-+-
Введите число одночленнов: 2
Введите число перед одночленном, а потом его степень: 2
2
3
2
-+-+-+Введем знаменатель+-+-+-
Введите число одночленнов: 2
Введите число перед одночленном, а потом его степень: 3
5
-3
1
Вы успешно ввели рациональную дробь!-+-+-+Введем числитель+-+-+-
Введите число одночленнов: 1
Введите число перед одночленном, а потом его степень: 2
2
-+-+-+Введем знаменатель+-+-+-
Введите число одночленнов: 2
Введите число перед одночленном, а потом его степень: 4
3
-4
3
Вы успешно ввели рациональную дробь!1 - Ввести рациональные дроби
2 - Вывести рациональные дроби
```

```
3 - Вывести результат сложения дробей
4 - Вывести результат перемножения дробей
5 - Завершить программу
Выберите действие ---- 2

************************************
2x^(2)+3x^(2)
------------------------------------
3x^(5)-3x
************************************
Вы успешно напечатали рациональную дробь!

************************************
2x^(2)
------------------------------------
4x^(3)-4x^(3)
************************************
Вы успешно напечатали рациональную дробь!
1 - Ввести рациональные дроби
2 - Вывести рациональные дроби
3 - Вывести результат сложения дробей
4 - Вывести результат перемножения дробей
5 - Завершить программу
Выберите действие ---- 3

************************************
Знаменатель обратился в нуль =(
------------------------------------
************************************
1 - Ввести рациональные дроби
2 - Вывести рациональные дроби
3 - Вывести результат сложения дробей
4 - Вывести результат перемножения дробей
5 - Завершить программу
Выберите действие ---- 4

************************************
Знаменатель обратился в нуль =(
------------------------------------
************************************
1 - Ввести рациональные дроби
2 - Вывести рациональные дроби
3 - Вывести результат сложения дробей
4 - Вывести результат перемножения дробей
5 - Завершить программу
Выберите действие ----
```

Тест 1

```
1 - Ввести рациональные дроби
2 - Вывести рациональные дроби
3 - Вывести результат сложения дробей
4 - Вывести результат перемножения дробей
5 - Завершить программу
Выберите действие ---- 1
-+-+-+Введем числитель+-+-+-
Введите число одночленнов: 2
Введите число перед одночленном, а потом его степень: 1 1 -9 0
-+-+-+Введем знаменатель+-+-+-
Введите число одночленнов: 2
Введите число перед одночленном, а потом его степень: 1 2 -9 0
Вы успешно ввели рациональную дробь!-+-+-+Введем числитель+-+-+-
Введите число одночленнов: 1
Введите число перед одночленном, а потом его степень: -3 0
-+-+-+Введем знаменатель+-+-+-
Введите число одночленнов: 2
Введите число перед одночленном, а потом его степень: 3 1 -1 2
Вы успешно ввели рациональную дробь!1 - Ввести рациональные дроби
2 - Вывести рациональные дроби
3 - Вывести результат сложения дробей
4 - Вывести результат перемножения дробей
5 - Завершить программу
Выберите действие ---- 2

************************************
1x-9
------------------------------------
1x^(2)-9
************************************
```

```
Вы успешно напечатали рациональную дробь!

************************************
-3
------------------------------------
3x-1x^(2)
************************************
Вы успешно напечатали рациональную дробь!
1 - Ввести рациональные дроби
2 - Вывести рациональные дроби
3 - Вывести результат сложения дробей
4 - Вывести результат перемножения дробей
5 - Завершить программу
Выберите действие ---- 3

************************************
9x^(2)-1x^(3)-27x+27
------------------------------------
3x^(3)-1x^(4)-27x+9x^(2)
************************************
Вы успешно напечатали рациональную дробь!
1 - Ввести рациональные дроби
2 - Вывести рациональные дроби
3 - Вывести результат сложения дробей
4 - Вывести результат перемножения дробей
5 - Завершить программу
Выберите действие ---- 4

************************************
-3x+27

------------------------------------
3x^(3)-1x^(4)-27x+9x^(2)
************************************
Вы успешно напечатали рациональную дробь!
1 - Ввести рациональные дроби
2 - Вывести рациональные дроби
3 - Вывести результат сложения дробей
4 - Вывести результат перемножения дробей
5 - Завершить программу
Выберите действие ----
```

Листинг

Main.java

```
package LW_7_TP;

import java.util.Scanner;
```

```
class Main {
   public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int number_of_action;
        Rational_fraction A = new Rational_fraction();
        Rational_fraction B = new Rational_fraction();
        Rational_fraction C;
        for (; ; ) {
            System.out.println("1 - Ввести рациональные дроби");
            System.out.println("2 - Вывести рациональные
дроби");
            System.out.println("3 - Вывести результат сложения
дробей");
            System.out.println("4 - Вывести результат
перемножения дробей");
            System.out.println("5 - Завершить программу");
            System.out.print("Выберите действие ---- ");
            number_of_action = scan.nextInt();
            switch (number_of_action) {
                case 1:
                    A.InputRationalFraction();
                    B.InputRationalFraction();
                    break;
                case 2:
                    A.PrintRationalFraction();
                    B.PrintRationalFraction();
                    break;
                case 3:
                    C = A.AdditionRationalFraction(B);
                    C.PrintRationalFraction();
                    break;
                case 4:
                    C = A.MultiplicationRationalFraction(B);
                    C.PrintRationalFraction();
                    break;
                case 5:
                    System.exit(1);
            }
        }
    }
}
```

Node_and_polinomial.java

```
package LW_7_TP;

import java.lang.System;

//Узел полинома
```

```
class Node {
    //значение коэффициента перед узлом
    int value;
    //степень узла
    int degree;
    //"указатель" на следующий узел
    Node next;
    //"указатель" на предыдущий узел
    Node prev;
}

class Polynom {
    Node first;
    Node last;

    public Polynom() {
        first = null;
        last = null;
    }

    public void AddNode(int v, int d) {
        Node p = new Node();
        if (first == null) {
            p.degree = d;
            p.value = v;
            p.prev = null;
            p.next = null;
            first = last = p;
        } else {
            p.value = v;
            p.degree = d;
            p.prev = last;
            p.next = null;
            last.next = p;
            last = p;
        }
    }

    public void WritePolynom() {
        if (first == null) {
            System.out.println("0");
        } else {
            Node p = first;
            while (p != null) {
                if (p != first) {
                    if (p.value > 0) {
                        System.out.print("+");
                    }
```

```
                }
                if (p.degree == 0) {
                    System.out.print(p.value);
                } else if (p.degree == 1) {
                    System.out.print(p.value);
                    System.out.print("x");
                } else {
                    System.out.print(p.value);
                    System.out.print("x^(");
                    System.out.print(p.degree);
                    System.out.print(")");
                }
                p = p.next;
            }
        }
    }

    public Polynom MultiplicationPolynom(Polynom S) {
        Polynom result = new Polynom();
        Node p = first;
        Node p_s;
        int sum_value;
        int sum_degree;
        while (p != null) {
            p_s = S.first;
            while (p_s != null) {
                sum_value = p.value * p_s.value;
                sum_degree = p.degree + p_s.degree;
                result.AddNode(sum_value, sum_degree);
                p_s = p_s.next;
            }
            p = p.next;
        }
        return result;
    }

    public void DeleteNode(Node p) {
        Node iteration_p = first;
        Node p_s;
        boolean flag = true;
        if (iteration_p == last) {
            first = null;
            last = null;
        } else {
            while (iteration_p != null && flag) {
                if (p == iteration_p) {
                    if (p == first) {
```

```
                    p_s = p.next;
                    p_s.prev = null;
                    first = p_s;
                    flag = false;
                } else if (p == last) {
                    p_s = p.prev;
                    p_s.next = null;
                    last = p_s;
                    flag = false;
                } else {
                    p_s = p.next;
                    p_s.prev = p.prev;
                    p.prev.next = p_s;
                    flag = false;
                }
            }
            iteration_p = iteration_p.next;
        }
    }
}
public boolean IsEmptyPolynom(){
    return this.first == null;
}

public Polynom SimplificationPolynom() {
    Polynom result = new Polynom();
    Node p = first;
    Node var_p;
    Node var_p_2;
    int sum;
    int current_degree;
    while (p != null) {
        current_degree = p.degree;
        sum = p.value;
        var_p = p.next;
        while (var_p != null) {
            if (current_degree == var_p.degree) {
                sum += var_p.value;
                var_p_2 = var_p.next;
                this.DeleteNode(var_p);
                var_p = var_p_2;
            } else
                var_p = var_p.next;
        }
        if (sum != 0) {
            result.AddNode(sum, current_degree);
        }
        p = p.next;
```

```
        }
        return result;
    }

    public Polynom AdditionPolynom(Polynom S) {
        Polynom result = new Polynom();
        Node p = first;
        Node p_s;
        Node p_s_2;
        int sum;
        while (p != null) {
            sum = p.value;
            p_s = S.first;
            while (p_s != null) {
                if (p.degree == p_s.degree) {
                    sum += p_s.value;
                    p_s_2 = p_s.next;
                    S.DeleteNode(p_s);
                    p_s = p_s_2;
                } else
                    p_s = p_s.next;
            }
            if (sum != 0)
                result.AddNode(sum, p.degree);
            p = p.next;
        }
        p = S.first;
        while (p != null) {
            result.AddNode(p.value, p.degree);
            p = p.next;
        }
        return result;
    }
}
```

Rational_fraction.java

```
package LW_7_TP;

import java.lang.System;
import java.util.Scanner;

public class Rational_fraction {
    Polynom numerator;
    Polynom denominator;

    public Rational_fraction() {
        numerator = new Polynom();
        denominator = new Polynom();
    }
```

```
    public void InputRationalFraction() {
        Scanner scan = new Scanner(System.in);
        System.out.println("-+-+-+Введем числитель+-+-+-");
        System.out.print("Введите число одночленнов: ");
        int num_of_members = scan.nextInt();
        System.out.print("Введите число перед одночленном, а
потом его степень: ");
        for (int i = 0; i < num_of_members; i++) {
            int value = scan.nextInt();
            int degree = scan.nextInt();
            numerator.AddNode(value, degree);
        }
        System.out.println("-+-+-+Введем знаменатель+-+-+-");
        System.out.print("Введите число одночленнов: ");
        num_of_members = scan.nextInt();
        System.out.print("Введите число перед одночленном, а
потом его степень: ");
        for (int i = 0; i < num_of_members; i++) {
            int value = scan.nextInt();
            int degree = scan.nextInt();
            denominator.AddNode(value, degree);
        }
        System.out.print("Вы успешно ввели рациональную
дробь!");
    }

    public void PrintRationalFraction() {
        System.out.println();

System.out.println("************************************");
        if (numerator.IsEmptyPolynom()) {
            System.out.print("0");

System.out.print("------------------------------------");
            System.out.println();

System.out.println("************************************");
        } else if (denominator.IsEmptyPolynom()) {
            System.out.println("Знаменатель обратился в нуль =(
");

System.out.print("------------------------------------");
            System.out.println();

System.out.println("************************************");
        } else {
            numerator.WritePolynom();
```

```
            System.out.println();

System.out.print("-------------------------------------");
            System.out.println();
            denominator.WritePolynom();
            System.out.println();

System.out.println("*************************************");
            System.out.println("Вы успешно напечатали
рациональную дробь!");
        }

    }

    public Rational_fraction
MultiplicationRationalFraction(Rational_fraction S) {
        Rational_fraction result = new Rational_fraction();
        this.denominator =
this.denominator.SimplificationPolynom();
        this.numerator = this.numerator.SimplificationPolynom();
        S.denominator = S.denominator.SimplificationPolynom();
        S.numerator = S.numerator.SimplificationPolynom();
        result.numerator =
this.numerator.MultiplicationPolynom(S.numerator).Simplificatio
nPolynom();
        result.denominator =
this.denominator.MultiplicationPolynom(S.denominator).Simplific
ationPolynom();
        return result;
    }

    public Rational_fraction
AdditionRationalFraction(Rational_fraction S) {
        Rational_fraction result = new Rational_fraction();
        this.denominator =
this.denominator.SimplificationPolynom();
        this.numerator = this.numerator.SimplificationPolynom();
        S.denominator = S.denominator.SimplificationPolynom();
        S.numerator = S.numerator.SimplificationPolynom();
        result.numerator =
this.numerator.MultiplicationPolynom(S.denominator).

SimplificationPolynom().AdditionPolynom(S.numerator.Multiplicat
ionPolynom(this.denominator).

SimplificationPolynom()).SimplificationPolynom();
```

```
        result.denominator =
this.denominator.MultiplicationPolynom(S.denominator).Simplific
ationPolynom(); return result; }}
```