



МАГИСТЕРСКАЯ ПРОГРАММА 09.04.01/07 Интеллектуальные системы анализа,
обработки и интерпретации больших данных

Вариант 6

Дисциплина: Языки программирования для работы с большими данными

Преподаватель	П.В. Степанов
_____ (Подпись, дата)	_____ (И.О. Фамилия)

Москва, 2022

Цель работы:

Получение навыков обработки исключений в Java и навыков работы с файлами в Java.

Выполнение:

Задание 1:

Выполнить задания на основе варианта 1 лабораторной работы 3, контролируя состояние потоков ввода/вывода. При возникновении ошибок, связанных с корректностью выполнения математических операций, генерировать и обрабатывать исключительные ситуации. Предусмотреть обработку исключений, возникающих при нехватке памяти, отсутствии требуемой записи (объекта) в файле, недопустимом значении поля и т.д.

1. Определить класс Матрица размерности (m x n). Класс должен содержать несколько конструкторов. Объявить массив объектов. Передать объекты в метод, меняющий местами строки с максимальным и минимальным элементами k-го столбца. Создать метод, который изменяет i-ю матрицу путем возведения ее в квадрат

2. Определить класс Цепная дробь

$$A = a_0 + \frac{x}{a_1 + \frac{x}{a_2 + \frac{x}{a_3 + \dots}}}$$

Определить методы сложения, вычитания, умножения, деления. Вычислить значение для заданного n, x, a[n].

Листинг выполнения подзадачи 1 (файл My_matrix.java)

```
package lr51;

import java.util.Arrays;

public class My_matrix {
    private int axis x;
```

```

private int axis_y;
private int[][] numbers = new int[axis_x][axis_y];
public My_matrix() {
}

public My_matrix(int axis_x, int axis_y, int[][] numbers) {
    this.axis_x = axis_x;
    this.axis_y = axis_y;
    this.numbers = numbers;
}

public My_matrix(int axis_x, int axis_y) {
    this.axis_x = axis_x;
    this.axis_y = axis_y;
    this.numbers = new int[axis_x][axis_y];
}

public void setAxis_x(int axis_x) {
    this.axis_x = axis_x;
}

public void setAxis_y(int axis_y) {
    this.axis_y = axis_y;
}

public int getAxis_x() {
    return axis_x;
}

public int getAxis_y() {
    return axis_y;
}

My_matrix square() {
    My_matrix new_m = new My_matrix(this.axis_x, this.axis_x);
    for (int i = 0; i < this.axis_x; i++) {
        for (int j = 0; j < this.axis_x; j++) {
            for (int k = 0; k < this.axis_y; k++) {
                new_m.numbers[i][j] += this.numbers[i][k] * this.numbers[j][k];
            }
        }
    }
    return new_m;
}

public int[][] getNumbers() {
    return numbers;
}

public void setNumbers(int[][] numbers) {
    this.numbers = numbers;
}

public void setNumber(int i, int j, int k) {
    this.numbers[i][j] = k;
}

public void change(int k) {
    int min = this.numbers[0][k];
    int max = this.numbers[0][k];
    int min_row = 0;
    int max_row = 0;
    for (int i = 0; i < this.axis_x; i++) {
        if (this.numbers[i][k] < min) {
            min = this.numbers[i][k];
            min_row = i;
        }
    }
}

```

```

        if (this.numbers[i][k] > max) {
            max = this.numbers[i][k];
            max_row = i;
        }
    }
    int[] mid = this.numbers[min_row];
    this.numbers[min_row] = this.numbers[max_row];
    this.numbers[max_row] = mid;
}

@Override
public String toString() {
    return "My_matrix{" +
        "axis_x=" + axis_x +
        ", axis_y=" + axis_y +
        ", numbers=" + Arrays.toString(numbers) +
        '}';
}
}

```

Листинг выполнения подзадачи 1 (файл lr515.java)

```

package lr515;

import java.util.Arrays;
import java.util.Scanner;
import java.util.concurrent.ThreadLocalRandom;

public class lr515 {
    public static void main(String[] args) {
        int amount = 1;
        Scanner in = new Scanner(System.in);
        try {
            amount = in.nextInt();
        } catch (Exception e) {
            System.out.println("baaad input, will be 1");
            in.nextLine();
        }

        My_matrix[] my_array = new My_matrix[amount];
        int n = 3 * amount;
        for (int i = 0; i < amount; i++) {
            try {
                int ax_x = ThreadLocalRandom.current().nextInt(2, n + 2);
                int ax_y = ThreadLocalRandom.current().nextInt(2, n + 2);
                my_array[i] = new My_matrix(ax_x, ax_y);
                for (int j = 0; j < ax_x; j++) {
                    for (int k = 0; k < ax_y; k++) {
                        my_array[i].setNumber(j, k,
                            ThreadLocalRandom.current().nextInt(-n, n + 1));
                    }
                }
            } catch (OutOfMemoryError e) {
                if (i==0) {
                    System.out.println("too many matrix, will be 1");
                    amount = 1;
                    n = 3;
                    i-=1;
                } else {
                    System.out.println("too many matrix, will be " + (i-1));
                    amount = i-1;
                    break;
                }
            }
        }
    }
}

```

```

for (int i = 0; i < amount; i++) {
    System.out.println("Matrix " + i);
    int[][] temp = my_array[i].getNumbers();
    for (int j = 0; j < temp.length; j++) {
        System.out.println(Arrays.toString(temp[j]));
    }
}
System.out.println("input k");
int for_swap = 0;
try{
    for_swap = in.nextInt();
} catch (Exception e){
    System.out.println("baaaad input, will be 0");
    in.nextLine();
}
for (int i = 0; i < amount; i++) {
    my_array[i].change(for_swap);
}
System.out.println("swapped lines:");
for (int i = 0; i < amount; i++) {
    System.out.println("Matrix " + i);
    int[][] temp = my_array[i].getNumbers();
    for (int j = 0; j < temp.length; j++) {
        System.out.println(Arrays.toString(temp[j]));
    }
}
System.out.println("input i");
int for_square = 0;
try{
    for_square = in.nextInt();
} catch (Exception e){
    System.out.println("baaaad input, will be zero");
    in.nextLine();
}
my_array[for_square] = my_array[for_square].square();
System.out.println("Squared matrix");
for (int i = 0; i < amount; i++) {
    System.out.println("Matrix " + i);
    int[][] temp = my_array[i].getNumbers();
    for (int j = 0; j < temp.length; j++) {
        System.out.println(Arrays.toString(temp[j]));
    }
}
}
}

```

3

Matrix 0

```
[-3, 8, 3, 2, 3, 8, -6, -1, -6, -9]
[9, 0, -2, 8, -2, -5, 7, -7, -8, -6]
[-4, 1, 4, -8, -8, 1, 7, -5, -5, -7]
[6, -5, 1, -3, 2, -4, -6, -5, 0, 9]
[-2, -2, -2, 0, -5, -3, 2, 4, -3, 6]
```

Matrix 1

```
[3, 5, 7, 8, 8, 8]
[-2, 7, 7, -6, 9, -7]
[9, 4, -2, -7, 4, -5]
```

Matrix 2

```
[0, 8, -2, 8]
[-4, -9, -7, 3]
[-8, -5, -4, 2]
[0, -7, -6, 5]
[-5, 8, 4, -2]
```

input k

1

swapped lines:

Matrix 0

```
[6, -5, 1, -3, 2, -4, -6, -5, 0, 9]
[9, 0, -2, 8, -2, -5, 7, -7, -8, -6]
[-4, 1, 4, -8, -8, 1, 7, -5, -5, -7]
[-3, 8, 3, 2, 3, 8, -6, -1, -6, -9]
[-2, -2, -2, 0, -5, -3, 2, 4, -3, 6]
```

Matrix 1

```
[3, 5, 7, 8, 8, 8]
[9, 4, -2, -7, 4, -5]
[-2, 7, 7, -6, 9, -7]
```

Matrix 2

```
[-4, -9, -7, 3]
[0, 8, -2, 8]
[-8, -5, -4, 2]
[0, -7, -6, 5]
[-5, 8, 4, -2]
```

```

input i
2
Squared matrix
Matrix 0
[6, -5, 1, -3, 2, -4, -6, -5, 0, 9]
[9, 0, -2, 8, -2, -5, 7, -7, -8, -6]
[-4, 1, 4, -8, -8, 1, 7, -5, -5, -7]
[-3, 8, 3, 2, 3, 8, -6, -1, -6, -9]
[-2, -2, -2, 0, -5, -3, 2, 4, -3, 6]
Matrix 1
[3, 5, 7, 8, 8, 8]
[9, 4, -2, -7, 4, -5]
[-2, 7, 7, -6, 9, -7]
Matrix 2
[155, -34, 111, 120, -86]
[-34, 132, -16, -4, 40]
[111, -16, 109, 69, -20]
[120, -4, 69, 110, -90]
[-86, 40, -20, -90, 109]

```

Рисунки 1,2 - Результат выполнения кода решения подзадачи 1

Листинг выполнения подзадачи 2 (файл Drob.java)

```

package lr51;

import java.util.Arrays;

public class Drob {
    private int amount;
    private int chisl;
    private int[] znam = new int[this.amount];

    public Drob(int amount, int chisl, int[] znam) {
        this.amount = amount;
        this.chisl = chisl;
        this.znam = znam;
    }

    public Drob() {
    }

    public Drob(int amount, int[] znam) {
        this.amount = amount;
        this.chisl = 1;
        this.znam = znam;
    }

    public int getAmount() {
        return amount;
    }

    public void setAmount(int amount) {
        this.amount = amount;
    }
}

```

```

public int getChisl() {
    return chisl;
}

public void setChisl(int chisl) {
    this.chisl = chisl;
}

public int[] getZnam() {
    return znam;
}

public void setZnam(int[] znam) {
    this.znam = znam;
}

double to_int(){
    double ans = 1;
    for (int i = this.amount - 1; i != 0; i--) {
        ans = this.znam[i] + 1/ans;
    }
    return ans;
}
static Drob from_int(double a){
    int[] new_znam = new int[100];
    int i = 1;
    new_znam[0] = (int)a;
    a-=new_znam[0];
    try {
        for (i = 1; i < 100; i++) {
            new_znam[i] = (int) (1 / a);
            a = 1 / a - new_znam[i];
            if (new_znam[i] == 0) {
                break;
            }
        }
    } catch (Exception e){
        System.out.println(e.getMessage());
        i = 0;
    }
    int[] new_new = new int[i];
    for (int j = 0; j < i; j++) {
        new_new[j] = new_znam[j];
    }
    return new Drob(i, 1, new_new);
}

public static Drob sum(Drob a, Drob b){
    return from_int(a.to_int() + b.to_int());
}
public static Drob min(Drob a, Drob b){
    return from_int(a.to_int() - b.to_int());
}
public static Drob mul(Drob a, Drob b){
    return from_int(a.to_int() * b.to_int());
}
public static Drob div(Drob a, Drob b){
    return from_int(a.to_int() / b.to_int());
}

@Override
public String toString() {
    return "Drob{" +
        "amount=" + amount +
        ", chisl=" + chisl +
        ", znam=" + Arrays.toString(znam) +

```



```

        '}'';
    }
}

```

Листинг выполнения подзадачи 2 (файл lr516.java)

```

package lr51;

import java.util.Scanner;
import java.util.concurrent.ThreadLocalRandom;

public class lr516 {
    public static void main(String[] args) {
        int amount = 1;
        try {
            Scanner in = new Scanner(System.in);
            amount = in.nextInt();
        } catch (Exception e) {
            System.out.println("baaaad input");
        }
        Drob[] my_array = new Drob[amount];
        int n = 3 * amount;
        for (int i = 0; i < amount; i++) {
            try {
                int a = ThreadLocalRandom.current().nextInt(2, n + 2);
                int b = ThreadLocalRandom.current().nextInt(2, n + 2);
                int[] c = new int[a];
                for (int j = 0; j < a; j++) {
                    c[j] = ThreadLocalRandom.current().nextInt(2, n + 1);
                }
                my_array[i] = new Drob(a, b, c);
            } catch (OutOfMemoryError e) {
                System.out.println("too many drobs, will be " + (i-1));
                amount = i-1;
                break;
            }
        }
        System.out.println("my_array:");
        for (int i = 0; i < amount; i++) {
            try {
                System.out.println(my_array[i]);
            } catch (OutOfMemoryError e) {
                System.out.println("too big to print");
            }
        }
        try {
            System.out.println("sum = " + Drob.sum(my_array[0], my_array[1]));
            System.out.println("min = " + Drob.min(my_array[0], my_array[1]));
            System.out.println("mul = " + Drob.mul(my_array[0], my_array[1]));
            System.out.println("div = " + Drob.div(my_array[0], my_array[1]));
        } catch (Exception e) {
            System.out.println("tooo less items");
        }
    }
}

```

```

C:\Users\alek\jdk\openjdk-17.0.2\bin\java.exe -javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\lib\idea_rt.jar=49751:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2
my_array:
Drob{amount=7, chisl=8, znam=[3, 12, 11, 11, 5, 4, 2]}
Drob{amount=10, chisl=6, znam=[3, 5, 11, 11, 5, 3, 6, 2, 5, 5]}
Drob{amount=3, chisl=3, znam=[6, 8, 12]}
Drob{amount=4, chisl=3, znam=[6, 6, 8, 12]}
sum = Drob{amount=100, chisl=1, znam=[17, 5, 1, 1, 5, 10, 1, 1, 4, 1, 4, 2, 1, 1, 1, 2, 33, 2, 8, 1, 3, 1, 5, 1, 11, 2, 124, 5, 1, 10, 12, 1, 1, 11, 1, 6, 7, 2, 5, 1, 2, 1, 1, 1, 1, 1, 49, 2, 2, 1, 1, 31, 6, 1, 2, 1, 1
min = Drob{amount=100, chisl=1, znam=[7, 4967933, 1, 3, 1, 125, 3, 1, 5, 1, 1, 49, 2, 1, 9, 1, 4, 1, 3, 3, 7, 85, 6, 2, 1, 4, 1, 4, 6, 1, 1, 2, 6, 7, 3, 26, 1, 4, 4, 3, 3, 4, 2, 14, 1, 59, 1, 12, 6, 16, 1, 1, 1, 1, 1,
mul = Drob{amount=100, chisl=1, znam=[61, 1, 1, 5, 1, 1, 2, 1, 2, 3, 8, 1, 1, 3, 4, 2, 9, 2, 1, 2, 1, 1, 45, 3, 1, 3, 6, 1, 1, 1, 3, 6, 1, 1, 2, 2, 50, 18, 3, 1, 1, 1, 6, 1, 3, 1, 2, 4, 4, 1, 83, 1, 3, 1, 1, 2, 34, 5,
div = Drob{amount=100, chisl=1, znam=[2, 2, 1, 1, 1, 78, 3, 3, 12, 1, 1, 1, 329, 2, 2, 1, 1, 105, 1, 2, 7, 1, 1, 1, 2, 1, 3, 1, 14, 1, 2, 2, 2, 47, 5, 1, 14, 48, 1, 10, 3, 1, 7, 1, 7, 2, 2, 1, 8, 2, 2, 1, 1, 3, 7, 2
Process finished with exit code 0

```

Рисунок 3 - Результат выполнения кода решения подзадачи 2

Задание 2:

Выполнить задания из варианта 2 лабораторной работы 3, реализуя собственные обработчики исключений и исключения ввода/вывода.

1. Book: id, Название, Автор(ы), Издательство, Год издания, Количество страниц, Цена, Переплет. Создать массив объектов. Вывести: а) список книг заданного автора; б) список книг, выпущенных заданным издательством; с) список книг, выпущенных после заданного года.
2. House: id, Номер квартиры, Площадь, Этаж, Количество комнат, Улица, Тип здания, Срок эксплуатации. Создать массив объектов. Вывести: а) список квартир, имеющих заданное число комнат; б) список квартир, имеющих заданное число комнат и расположенных на этаже, который находится в заданном промежутке; с) список квартир, имеющих площадь, превосходящую заданную.

Листинг выполнения подзадачи 1 (файл book.java)

```
package lr52;

public class Book {
    private int id;
    private String name;
    private String author;
    private String publisher;
    private int year;
    private int page_amount;
    private int cost;
    private boolean cover;

    public Book() {

    }

    public Book(int id, String name, String author, String publisher, int year, int
page_amount, int cost, boolean cover) throws Exception {
        this.id = id;
        this.name = name;
        this.author = author;
        this.publisher = publisher;
        this.year = year;
```

```

        if (page_amount < 0){
            throw new Exception("Page_amount < 0");
        }
        this.page_amount = page_amount;
        if (cost < 0){
            throw new Exception("Cost < 0");
        }
        this.cost = cost;
        this.cover = cover;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getAuthor() {
        return author;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    public String getPublisher() {
        return publisher;
    }

    public void setPublisher(String publisher) {
        this.publisher = publisher;
    }

    public int getYear() {
        return year;
    }

    public void setYear(int year) {
        this.year = year;
    }

    public int getPage_amount() {
        return page_amount;
    }

    public void setPage_amount(int page_amount) {
        this.page_amount = page_amount;
    }

    public int getCost() {
        return cost;
    }

    public void setCost(int cost) {
        this.cost = cost;
    }

```

```

    }

    public boolean isCover() {
        return cover;
    }

    public void setCover(boolean cover) {
        this.cover = cover;
    }

    public static Book[] make_list(int amount) {
        return new Book[amount];
    }

    @Override
    public String toString() {
        return "Book{" +
            "id=" + id +
            ", name='" + name + '\'' +
            ", author='" + author + '\'' +
            ", publisher='" + publisher + '\'' +
            ", year=" + year +
            ", page_amount=" + page_amount +
            ", cost=" + cost +
            ", cover=" + cover +
            '}';
    }
}

```

Листинг выполнения подзадачи 1 (файл lr525.java)

```

package lr52;

import java.util.Objects;

public class lr525 {
    public static void main(String[] args) {
        Book[] my_book = Book.make_list(5);
        try {
            my_book[0] = new Book(1, "1", "1", "1", 1, 1, 1, false);
        } catch (Exception e) {
            e.printStackTrace();
        }
        try {
            my_book[1] = new Book(2, "2", "1", "1", 2, 1, 1, false);
        } catch (Exception e) {
            e.printStackTrace();
        }
        try {
            my_book[2] = new Book(3, "3", "2", "1", 3, 1, -1, false);
        } catch (Exception e) {
            e.printStackTrace();
        }
        try {
            my_book[3] = new Book(4, "4", "2", "1", 4, -1, 1, false);
        } catch (Exception e) {
            e.printStackTrace();
        }
        try {
            my_book[4] = new Book(5, "5", "2", "2", 5, 1, 1, false);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

```

        System.out.println("books of author 2:");
        for (int i = 0; i < 5; i++) {
            try {
                if (Objects.equals(my_book[i].getAuthor(), "2")) {
                    System.out.println(my_book[i]);
                }
            } catch (Exception ignored) {}
        }
        System.out.println("books of publisher 1:");
        for (int i = 0; i < 5; i++) {
            try {
                if (Objects.equals(my_book[i].getPublisher(), "1")) {
                    System.out.println(my_book[i]);
                }
            } catch (Exception ignored) {}
        }
        System.out.println("books of year later 3:");
        for (int i = 0; i < 5; i++) {
            try {
                if (my_book[i].getYear() > 3) {
                    System.out.println(my_book[i]);
                }
            } catch (Exception ignored) {}
        }
    }
}

```

```

C:\Users\stale\.jdk\openjdk-17.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\Intelli
java.lang.Exception Create breakpoint : Cost < 0
    at lr52.Book.<init>(Book.java:28)
    at lr52.lr525.main(lr525.java:19)
java.lang.Exception Create breakpoint : Page_amount < 0
    at lr52.Book.<init>(Book.java:24)
    at lr52.lr525.main(lr525.java:24)
books of author 2:
Book{id=5, name='5', author='2', publisher='2', year=5, page_amount=1, cost=1, cover=false}
books of publisher 1:
Book{id=1, name='1', author='1', publisher='1', year=1, page_amount=1, cost=1, cover=false}
Book{id=2, name='2', author='1', publisher='1', year=2, page_amount=1, cost=1, cover=false}
books of year later 3:
Book{id=5, name='5', author='2', publisher='2', year=5, page_amount=1, cost=1, cover=false}

Process finished with exit code 0

```

Рисунок 4 - Результат выполнения кода решения подзадачи 1

Листинг выполнения подзадачи 2 (файл House.java)

```

package lr52;

public class House {
    private int id;
    private int house_number;
    private int square;
    private int floor;
    private int room_amount;
}

```

```

private String street;
private boolean type;
private int availability_till;

public House(int id, int house_number, int square, int floor, int room_amount,
String street, boolean type, int availability_till) throws Exception {
    this.id = id;
    this.house_number = house_number;
    this.square = square;
    if (floor < 1){
        throw new Exception("Floor < 1");
    }
    this.floor = floor;
    if (room_amount < 1){
        throw new Exception("Room_amount < 1");
    }
    this.room_amount = room_amount;
    this.street = street;
    this.type = type;
    this.availability_till = availability_till;
}

public House() {
}

public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

public int getHouse_number() {
    return house_number;
}

public void setHouse_number(int house_number) {
    this.house_number = house_number;
}

public int getSquare() {
    return square;
}

public void setSquare(int square) {
    this.square = square;
}

public int getFloor() {
    return floor;
}

public void setFloor(int floor) {
    this.floor = floor;
}

public int getRoom_amount() {
    return room_amount;
}

public void setRoom_amount(int room_amount) {
    this.room_amount = room_amount;
}

public String getStreet() {

```

```

        return street;
    }

    public void setStreet(String street) {
        this.street = street;
    }

    public boolean isType() {
        return type;
    }

    public void setType(boolean type) {
        this.type = type;
    }

    public int getAvailability_till() {
        return availability_till;
    }

    public void setAvailability_till(int availability_till) {
        this.availability_till = availability_till;
    }

    public static House[] make_list(int amount){
        return new House[amount];
    }

    @Override
    public String toString() {
        return "House{" +
            "id=" + id +
            ", house_number=" + house_number +
            ", square=" + square +
            ", floor=" + floor +
            ", room_amount=" + room_amount +
            ", street='" + street + '\'' +
            ", type=" + type +
            ", availability_till=" + availability_till +
            '}';
    }
}

```

Листинг выполнения подзадачи 2 (файл lr526.java)

```

package lr52;

public class lr526 {
    public static void main(String[] args){
        House[] my_house = House.make_list(5);

        try {
            my_house[0] = new House(1, 1, 1, 1, 1, "1", false, 1);
        } catch (Exception e) {
            e.printStackTrace();
        }
        try {
            my_house[1] = new House(2, 1, 2, 2, 1, "1", false, 1);
        } catch (Exception e) {
            e.printStackTrace();
        }
        try {
            my_house[2] = new House(3, 1, 3, 1, 2, "1", false, 1);
        } catch (Exception e) {
            e.printStackTrace();
        }
        try {

```

```

        my_house[3] = new House(4, 1, 4, 3, -1, "1", false, 1);
    } catch (Exception e) {
        e.printStackTrace();
    }
    try {
        my_house[4] = new House(5, 1, 5, -1, 1, "1", false, 1);
    } catch (Exception e) {
        e.printStackTrace();
    }
    System.out.println("flats with 1 room:");
    for (int i = 0; i < 5; i++) {
        try {
            if (my_house[i].getRoom_amount() == 1) {
                System.out.println(my_house[i]);
            }
        } catch (Exception ignored){}
    }
    System.out.println("flats with 1 room between 2 and 3 floors:");
    for (int i = 0; i < 5; i++) {
        try {
            if (my_house[i].getRoom_amount() == 1 && my_house[i].getFloor() >= 2
&& my_house[i].getFloor() <= 3) {
                System.out.println(my_house[i]);
            }
        } catch (Exception ignored){}
    }
    System.out.println("flats Square > 2");
    for (int i = 0; i < 5; i++) {
        try {
            if (my_house[i].getSquare() > 2) {
                System.out.println(my_house[i]);
            }
        } catch (Exception ignored){}
    }
}
}

```

```

C:\Users\stale\.jdk\openjdk-17.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Commur
java.lang.Exception Create breakpoint : Room_amount < 1
    at lr52.House.<init>(House.java:22)
    at lr52.Lr526.main(Lr526.java:23)
java.lang.Exception Create breakpoint : Floor < 1
    at lr52.House.<init>(House.java:18)
    at lr52.Lr526.main(Lr526.java:28)
flats with 1 room:
House{id=1, house_number=1, square=1, floor=1, room_amount=1, street='1', type=false, availability_till=1}
House{id=2, house_number=1, square=2, floor=2, room_amount=1, street='1', type=false, availability_till=1}
flats with 1 room between 2 and 3 floors:
House{id=2, house_number=1, square=2, floor=2, room_amount=1, street='1', type=false, availability_till=1}
flats Square > 2
House{id=3, house_number=1, square=3, floor=1, room_amount=2, street='1', type=false, availability_till=1}
Process finished with exit code 0

```

Рисунок 5 - Результат выполнения кода решения подзадачи 2

Задание 3:

В следующих заданиях требуется ввести последовательность строк из текстового потока и выполнить указанные действия. При этом могут рассматриваться два варианта:

- Найти в строке наибольшее число цифр, идущих подряд;
- В каждой строке стихотворения Анны Ахматовой подсчитать частоту повторяемости каждого слова из заданного списка и вывести эти слова в порядке возрастания частоты повторяемости..

Имена входного и выходного файлов, а также абсолютный путь к ним могут быть введены как параметры командной строки или храниться в файле.

1. В каждой строке стихотворения Александра Блока найти и заменить заданную подстроку на подстроку иной длины.
2. В каждой строке найти слова, начинающиеся с гласной буквы.

Текстовый файл для выполнения подзадачи 1 (файл lr535_in.fts)

```
asdasdaff12233 1
12222222222222 a
asdasa sdsad
1222 12
```

Листинг выполнения подзадачи 1 (файл lr535.java)

```
package lr53;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.nio.file.StandardOpenOption;
import java.util.Collections;
import java.util.Scanner;

public class lr535 {
    public static void main(String[] args) throws IOException {
        File file_in = new
File("C:\\Users\\stale\\IdeaProjects\\java_magister\\lr5\\src\\lr53\\lr535_in.fts");
        Path path =
Paths.get("C:\\Users\\stale\\IdeaProjects\\java_magister\\lr5\\src\\lr53\\lr535_out.f
ts");
        Scanner in = new Scanner(file_in);
        String line;
        boolean first = true;
        while(in.hasNextLine()) {
            line = in.nextLine();
```

```

        int max = 0;
        int curr = 0;
        for (int i = 0; i < line.length(); i++) {
            if ('0' <= line.charAt(i) && line.charAt(i) <= '9'){
                curr++;
                if (curr > max){
                    max = curr;
                }
            } else {
                curr = 0;
            }
        }
        if (first) {
            Files.write(path, Collections.singleton(String.valueOf(max)),
StandardCharsets.UTF_8);
            first = false;
        } else {
            Files.write(path, Collections.singleton(String.valueOf(max)),
StandardCharsets.UTF_8, StandardOpenOption.APPEND);
        }
    }
}
}

```

Результирующий текстовый файл после выполнения подзадачи 1 (файл lr535_out.fts)

```

5
13
0
4

```

Текстовый файл для выполнения подзадачи 2 (файл lr536_in.fts)

```

Перед весной бывают дни такие:
Под плотным снегом отдыхает луг,
Шумят деревья весело-сухие,
И тёплый ветер нежен и упруг.
И лёгкости своей дивится тело,
И дома своего не и узнаешь,
И песню ту, что прежде надоела,
Как новую, с волнением поешь.

```

Листинг выполнения подзадачи 2 (файл lr536.java)

```

package lr53;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.nio.file.StandardOpenOption;
import java.util.*;

public class lr536 {
    public static void main(String[] args) throws IOException {
        File file_in = new
File("C:\\Users\\stale\\IdeaProjects\\java_magister\\lr5\\src\\lr53\\lr536_in.fts");
        Path path =
Paths.get("C:\\Users\\stale\\IdeaProjects\\java_magister\\lr5\\src\\lr53\\lr536_out.f
ts");
        Scanner in = new Scanner(file_in);
        String line;
        HashMap<String, Integer> key_words = filling();
    }
}

```

```

        int k = 1;
        while(in.hasNextLine()) {
            line = in.nextLine();
            String[] words = line.split(" ");
            for (String i : words) {
                if (key_words.containsKey(i.toLowerCase(Locale.ROOT))) {
                    key_words.put(i.toLowerCase(Locale.ROOT),
key_words.get(i.toLowerCase(Locale.ROOT)) + 1);
                }
            }
            String answer = new String();
            answer += ("line " + k + ": ");
            for (String i: key_words.keySet()) {
                int max = 0;
                String max2 = null;
                for (String j: key_words.keySet()) {
                    if (key_words.get(j) > max) {
                        max2 = j;
                        max = key_words.get(j);
                    }
                }
                try {
                    if (key_words.get(max2) != 0) {
                        answer += max2 + ", ";
                    }
                    key_words.put(max2, 0);
                } catch (Exception ignored) {}
            }
            if (k == 1) {
                Files.write(path, Collections.singleton(answer),
StandardCharsets.UTF_8);
            } else {
                Files.write(path, Collections.singleton(answer),
StandardCharsets.UTF_8, StandardOpenOption.APPEND);
            }
            key_words = filling();
            k++;
        }

        public static HashMap filling(){
            HashMap<String, Integer> a = new HashMap<>();
            a.put("не", 0);
            a.put("и", 0);
            a.put("как", 0);
            return a;
        }
    }
}

```

Результирующий текстовый файл после выполнения подзадачи 2 (файл lr536_out.fts)

```

line 1:
line 2:
line 3:
line 4: и,
line 5: и,
line 6: и, не,
line 7: и,
line 8: как,

```

Задание 4:

При выполнении следующих заданий для вывода результатов создавать новую директорию и файл средствами класса File.

1. Файл содержит символы, слова, целые числа и числа с плавающей запятой. Определить все данные, тип которых вводится из командной строки.
2. Из файла удалить все слова, содержащие от трех до пяти символов, но при этом из каждой строки должно быть удалено только максимальное четное количество таких слов.

Текстовый файл для выполнения подзадачи 1 (файл lr545_in.fts)

```
aaaaa
a
12
12.1
```

Листинг выполнения подзадачи 1 (файл lr545.java)

```
package lr54;

import java.io.File;
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.nio.file.StandardOpenOption;
import java.util.Collections;
import java.util.Scanner;
import java.util.regex.Pattern;

public class lr545 {
    public static void main(String[] args) throws IOException {
        File file_in = new
File("C:\\Users\\stale\\IdeaProjects\\java_magister\\lr5\\src\\lr54\\lr545_in.fts");
        File file_out = new
File("C:\\Users\\stale\\IdeaProjects\\java_magister\\lr5\\src\\lr54\\lr545_out.fts");
        file_out.delete();
        file_out.createNewFile();
        Path path = Path.of(file_out.getPath());
        Scanner in = new Scanner(file_in);
        String line;
        Scanner in_cmd = new Scanner(System.in);
        int choose = in_cmd.nextInt();
        while(in.hasNextLine()) {
            line = in.next();
            if (choose == 1 && Pattern.matches("[0-9]+", line)) { //Число
                Files.write(path, Collections.singleton(line),
StandardCharsets.UTF_8, StandardOpenOption.APPEND);
            }
        }
    }
}
```

```

    }
    if (choose == 2 && Pattern.matches("[+-
]?(?:=\\.\\.d|\\.d)(?:\\.d+)?(?:\\.?.\\.d*)?(?:[eE]([+-]?\\.d+)?", line)) { //float
        Files.write(path, Collections.singleton(line),
StandardCharsets.UTF_8, StandardOpenOption.APPEND);
    }
    if (choose == 3 && Pattern.matches("[a-zA-Z]", line)) { //char
        Files.write(path, Collections.singleton(line),
StandardCharsets.UTF_8, StandardOpenOption.APPEND);
    }
    if (choose == 4 && Pattern.matches("[a-zA-Z]+", line)) { //string
        Files.write(path, Collections.singleton(line),
StandardCharsets.UTF_8, StandardOpenOption.APPEND);
    }
}
}
}

```

Результирующий текстовый файл после выполнения подзадачи 1 (файл lr545_out.fts)

```
12
```

Текстовый файл для выполнения подзадачи 2 (файл lr546_in.fts)

```

aaaa bbbb ssss
aaaa aaaaaaaaaa
aaaa aaaaaaaaaa aaaa
aaaa aaaaaaaaaa aa
aaaa aaaa s aaaa aaaaa

```

Листинг выполнения подзадачи 2 (файл lr536.java)

```

package lr54;

import java.io.File;
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.StandardOpenOption;
import java.util.*;
import java.util.regex.Pattern;

public class lr546 {
    public static void main(String[] args) throws IOException {
        File file_in = new
File("C:\\Users\\stale\\IdeaProjects\\java_magister\\lr5\\src\\lr54\\lr546_in.fts");
        File file_out = new
File("C:\\Users\\stale\\IdeaProjects\\java_magister\\lr5\\src\\lr54\\lr546_out.fts");
        file_out.delete();
        file_out.createNewFile();
        Path path = Path.of(file_out.getPath());
        Scanner in = new Scanner(file_in);
        String line;
        while(in.hasNextLine()) {
            line = in.nextLine();
            ArrayList<String> words = new ArrayList<>(List.of(line.split(" ")));
            int i = 0;
            int first = -1;
            while (i < words.size()) {
                if (3 <= words.get(i).length() && words.get(i).length() <= 5) {
                    if (first == -1) {
                        first = i;
                        i++;
                    }
                }
            }
            else{

```

```

        words.remove(i);
        words.remove(first);
        i--;
        first = -1;
    }
    }else {
        i++;
    }
}
String answer = "";
for (int j = 0; j < words.size(); j++) {
    answer += words.get(j) + " ";
}
Files.write(path, Collections.singleton(answer), StandardCharsets.UTF_8,
StandardOpenOption.APPEND);
    }
}
}

```

Результирующий текстовый файл после выполнения подзадачи 2 (файл lr536_out.fts)

```

ssss
aaaa aaaaaaaaaa
aaaaaaaaaa
aaaa aaaaaaaaaa aa
s

```

Вывод:

При выполнении лабораторной работы были получены навыки обработки исключений в Java и навыки работы с файлами в Java.