



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 КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.04.01 Информатика и вычислительная техника

МАГИСТЕРСКАЯ ПРОГРАММА 09.04.01/12 Интеллектуальный анализ больших  
данных в системах поддержки принятия решений.

**О Т Ч Е Т**

**по лабораторной работе № 3**

**Вариант № 4**

**Название:** классы, наследование и полиморфизм

**Дисциплина:** языки программирования для работы с большими данными

Студент

ИУ6-23М

(Группа)

(Подпись, дата)

А.А.Клушина

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

П.В. Степанов

(И.О. Фамилия)

Москва, 2024

**Цель:** освоить принципы ООП на языке программирования Java.

**Задание 1:** Определить класс Матрица размерности (n x n). Класс должен содержать несколько конструкторов. Реализовать методы для сложения, вычитания, умножения матриц. Объявить массив объектов. Создать методы, вычисляющие первую и вторую нормы матрицы.

$$\|a\|_1 = \max_{1 \leq i \leq n} \sum_{j=1}^n (a_{ij}), \|a\|_2 = \max_{1 \leq j \leq n} \sum_{i=1}^n (a_{ij})$$

Код класса Main:

```
public class Matrix {
    private int[][] matrix;
    private int size;

    public Matrix(int size) {
        this.size = size;
        this.matrix = new int[size][size];
    }

    public Matrix(int size, int[][] matrix) {
        this.size = size;
        this.matrix = matrix;
    }

    public Matrix add(Matrix other) {
        int[][] result = new int[size][size];
        for (int i = 0; i < size; i++) {
            for (int j = 0; j < size; j++) {
                result[i][j] = this.matrix[i][j] +
other.matrix[i][j];
            }
        }
        return new Matrix(size, result);
    }

    public Matrix subtract(Matrix other) {
        int[][] result = new int[size][size];
        for (int i = 0; i < size; i++) {
            for (int j = 0; j < size; j++) {
                result[i][j] = this.matrix[i][j] -
other.matrix[i][j];
            }
        }
        return new Matrix(size, result);
    }

    public Matrix multiply(Matrix other) {
        int[][] result = new int[size][size];
```

```

        for (int i = 0; i < size; i++) {
            for (int j = 0; j < size; j++) {
                int sum = 0;
                for (int k = 0; k < size; k++) {
                    sum += this.matrix[i][k] *
other.matrix[k][j];
                }
                result[i][j] = sum;
            }
        }
        return new Matrix(size, result);
    }

    public double calculateFirstNorm() {
        double norm = 0;
        for (int i = 0; i < size; i++) {
            double sum = 0;
            for (int j = 0; j < size; j++) {
                sum += Math.abs(matrix[i][j]);
            }
            norm = Math.max(norm, sum);
        }
        return norm;
    }

    public double calculateSecondNorm() {
        double norm = 0;
        for (int i = 0; i < size; i++) {
            double sum = 0;
            for (int j = 0; j < size; j++) {
                sum += Math.abs(matrix[j][i]);
            }
            norm = Math.max(norm, sum);
        }
        return norm;
    }

    public static void printMatrix(Matrix matrix) {
        System.out.println("Matrix:");
        for (int i = 0; i < matrix.size; i++) {
            for (int j = 0; j < matrix.size; j++) {
                System.out.print(matrix.matrix[i][j] + " ");
            }
            System.out.println();
        }
    }

    public static void main(String[] args) {
        Matrix[] matrices = new Matrix[3]; // Создаем массив
        объектов

        // Создаем и инициализируем матрицы
        int[][] matrix1 = {{1, 2}, {3, 4}};

```

```

matrices[0] = new Matrix(2, matrix1);

int[][] matrix2 = {{5, 6}, {7, 8}};
matrices[1] = new Matrix(2, matrix2);

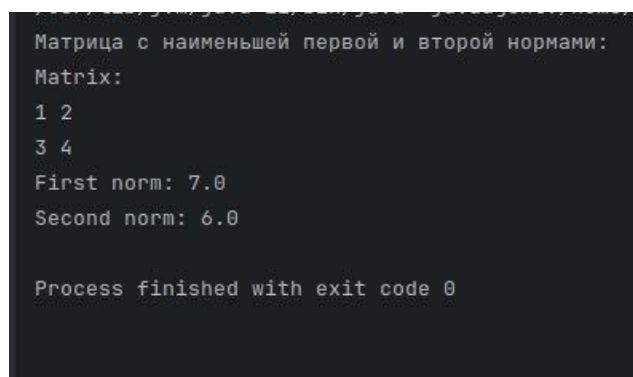
int[][] matrix3 = {{-1, -2}, {-3, -4}};
matrices[2] = new Matrix(2, matrix3);

// Находим матрицу с наименьшей первой и второй нормами
Matrix minMatrix = matrices[0];
for (int i = 1; i < matrices.length; i++) {
    if (matrices[i].calculateFirstNorm() <
minMatrix.calculateFirstNorm() &&
        matrices[i].calculateSecondNorm() <
minMatrix.calculateSecondNorm()) {
        minMatrix = matrices[i];
    }
}

System.out.println("Матрица с наименьшей первой и второй
нормами:");
printMatrix(minMatrix);
System.out.println("First norm: " +
minMatrix.calculateFirstNorm());
System.out.println("Second norm: " +
minMatrix.calculateSecondNorm());
}
}

```

Работа программы показана на рисунке 1.



```

Матрица с наименьшей первой и второй нормами:
Matrix:
1 2
3 4
First norm: 7.0
Second norm: 6.0

Process finished with exit code 0

```

Рисунок 1 – Работа программы

## Задание 2:

Определить класс Матрица размерности ( $m \times n$ ). 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.

### Код класса Main:

```
import java.util.Scanner;

class Matrix {
    private int rows;
    private int cols;
    private int[][] data;

    public Matrix(int rows, int cols) {
        this.rows = rows;
        this.cols = cols;
        this.data = new int[rows][cols];
    }

    public Matrix() {
        this(2, 2);
    }

    public void fill() {
        Scanner input = new Scanner(System.in);
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                System.out.print("Введите элемент матрицы (" + i
+ ", " + j + "): ");
                data[i][j] = input.nextInt();
            }
        }
    }

    public void printMatrix() {
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                System.out.print(data[i][j] + " ");
            }
            System.out.println();
        }
    }

    public void swapRowsWithMinMax(int k) {
        int minElement = data[0][k];
        int maxElement = data[0][k];
        int minRow = 0;
        int maxRow = 0;

        for (int i = 1; i < rows; i++) {
            if (data[i][k] < minElement) {
                minElement = data[i][k];
                minRow = i;
            }
            if (data[i][k] > maxElement) {
                maxElement = data[i][k];
                maxRow = i;
            }
        }
    }
}
```

```

        }
    }

    int[] temp = data[minRow];
    data[minRow] = data[maxRow];
    data[maxRow] = temp;
}

public void squareMatrix() {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            data[i][j] *= data[i][j];
        }
    }
}

public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    System.out.print("Введите количество строк матрицы: ");
    int rows = input.nextInt();
    System.out.print("Введите количество столбцов матрицы: ");
    int cols = input.nextInt();

    Matrix matrix = new Matrix(rows, cols);

    matrix.fill();
    System.out.println("Матрица:");
    matrix.printMatrix();
    System.out.print("Введите столбец k: ");
    int k = input.nextInt();
    matrix.swapRowsWithMinMax(k);
    System.out.println("Матрица после перемены строк:");
    matrix.printMatrix();
    matrix.squareMatrix();
    System.out.println("Матрица после возведения в
квадрат:");
    matrix.printMatrix();
}
}

```

**Работа программы показана на рисунке 2.**

```
Введите количество строк матрицы: 3
Введите количество столбцов матрицы: 3
Введите элемент матрицы (0, 0): 1
Введите элемент матрицы (0, 1): 2
Введите элемент матрицы (0, 2): 3
Введите элемент матрицы (1, 0): 4
Введите элемент матрицы (1, 1): 5
Введите элемент матрицы (1, 2): 6
Введите элемент матрицы (2, 0): 7
Введите элемент матрицы (2, 1): 8
Введите элемент матрицы (2, 2): 8
Матрица:
1 2 3
4 5 6
7 8 8
Введите столбец k: 2
Матрица после перемены строк:
7 8 8
4 5 6
1 2 3
Матрица после возведения в квадрат:
49 64 64
16 25 36
1 4 9
```

Рисунок 2 – Работа программы

**Задание 3:** Создать классы, спецификации которых приведены ниже. Определить конструкторы и методы `setТип()`, `getТип()`, `toString()`. Определить дополнительно методы в классе, создающем массив объектов. Задать критерий выбора данных и вывести эти данные на консоль. `Abiturient`: `id`, `Фамилия`, `Имя`, `Отчество`, `Адрес`, `Телефон`, `Оценки`. Создать массив объектов. Вывести: а) список абитуриентов, имеющих неудовлетворительные оценки; б) список абитуриентов, средний балл у которых выше заданного; с) выбрать заданное число `n` абитуриентов, имеющих самый высокий средний балл (вывести также полный список абитуриентов, имеющих полупроходной балл).

Код класса `Abiturient`:

```
import java.util.Scanner;
import java.util.ArrayList;

class Abiturient {
    private int id;
    private String surname;
    private String firstName;
```

```

private String middleName;
private String address;
private String phoneNumber;
private int[] grades;

public Abiturient(int id, String surname, String firstName,
String middleName, String address, String phoneNumber, int[]
grades) {
    this.id = id;
    this.surname = surname;
    this.firstName = firstName;
    this.middleName = middleName;
    this.address = address;
    this.phoneNumber = phoneNumber;
    this.grades = grades;
}

public double calculateAverageGrade() {
    int sum = 0;
    for (int grade : grades) {
        sum += grade;
    }
    return (double) sum / grades.length;
}

public int getId() {
    return id;
}

public String getFullName() {
    return surname + " " + firstName + " " + middleName;
}

public int[] getGrades() {
    return grades;
}

public static void main(String[] args) {
    Scanner input = new Scanner(System.in);

    ArrayList<Abiturient> abiturients = new ArrayList<>();
    abiturients.add(new Abiturient(1, "Чернавина", "Ксения",
"Максимовна", "Ярославль", "123456789", new int[]{80, 70, 60}));
    abiturients.add(new Abiturient(2, "Клушина",
"Анастасия", "Алексеевна", "Москва", "987654321", new int[]{90,
85, 75}));
    abiturients.add(new Abiturient(3, "Тюкин", "Иван",
"Дмитриевич", "Владивосток", "555555555", new int[]{50, 55,
70}));

    // а) список абитуриентов, имеющих неудовлетворительные
оценки
    System.out.println("Список абитуриентов, имеющих
неудовлетворительные оценки:");
}

```



```

        for (Abiturient abiturient : abiturients) {
            for (int grade : abiturient.getGrades()) {
                if (grade < 60) {

System.out.println(abiturient.getFullName());
                    break;
                }
            }
        }

        // b) список абитуриентов, средний балл у которых выше
заданного
        System.out.print("Введите балл: ");
        double minAverageGrade = input.nextDouble();
        System.out.println("Список абитуриентов, средний балл у
которых выше " + minAverageGrade + ":");
        for (Abiturient abiturient : abiturients) {
            if (abiturient.calculateAverageGrade() >
minAverageGrade) {
                System.out.println(abiturient.getFullName());
            }
        }

        // c) выбрать заданное число n абитуриентов, имеющих
самый высокий средний балл
        System.out.print("Введите количество абитуриентов: ");
        int n = input.nextInt();
        abiturients.sort((a, b) ->
Double.compare(b.calculateAverageGrade(),
a.calculateAverageGrade()));
        System.out.println("Топ " + n + " абитуриентов с
максимальным средним баллом:");
        for (int i = 0; i < n; i++) {

System.out.println(abiturients.get(i).getFullName());

        }

        // полный список абитуриентов с полупроходным баллом
        double passingGrade = 60.0;
        System.out.println("Список абитуриентов с полупроходным
баллом:");
        for (int i = n; i < abiturients.size(); i++) {
            if (abiturients.get(i).calculateAverageGrade() >=
passingGrade) {

System.out.println(abiturients.get(i).getFullName());
                }
            }
        }
    }
}

```

Работа программы показана на рисунке 3.

```
Список абитуриентов, имеющих неудовлетворительные оценки:
Тюкин Иван Дмитриевич
Введите балл: 12
Список абитуриентов, средний балл у которых выше 12.0:
Чернавина Ксения Максимовна
Клушина Анастасия Алексеевна
Тюкин Иван Дмитриевич
Введите количество абитуриентов: 2
Топ 2 абитуриентов с максимальным средним баллом:
Клушина Анастасия Алексеевна
Чернавина Ксения Максимовна
Список абитуриентов с полупроходным баллом:
```

Рисунок 3 – Работа программы

**Задание 4:** создать классы, спецификации которых приведены ниже. Определить конструкторы и методы `setТип()`, `getТип()`, `toString()`. Определить дополнительно методы в классе, создающем массив объектов. Задать критерий выбора данных и вывести эти данные на консоль. 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.

Код класса Book:

```
import java.util.Scanner;

class Book {
    private int id;
    private String title;
    private String author;
    private String publisher;
    private int year;
    private int pageCount;
    private double price;
    private String binding;

    public Book(int id, String title, String author, String publisher, int year, int pageCount, double price, String binding) {
        this.id = id;
        this.title = title;
        this.author = author;
```

```

        this.publisher = publisher;
        this.year = year;
        this.pageCount = pageCount;
        this.price = price;
        this.binding = binding;
    }

    public String getAuthor() {
        return author;
    }

    public String getPublisher() {
        return publisher;
    }

    public int getYear() {
        return year;
    }

    public void displayInfo() {
        System.out.println("Название: " + title);
        System.out.println("Автор(ы): " + author);
        System.out.println("Издательство: " + publisher);
        System.out.println("Год: " + year);
        System.out.println("Количество страниц: " + pageCount);
        System.out.println("Цена: " + price + " руб.");
        System.out.println("Переплет: " + binding);
        System.out.println();
    }

    public static void main(String[] args) {
        Book[] books = {
            new Book(1, "Преступление и наказание",
"Достоевский", "Москва", 2000, 300, 20.0, "Твердый"),
            new Book(2, "Дракула", "Стокер", "Издательство",
2005, 250, 15.0, "Мягкий"),
            new Book(3, "1984", "Оруэлл", "АСТ", 2010, 400,
30.0, "Твердый")
        };

        Scanner scanner = new Scanner(System.in);

        // a) список книг заданного автора
        System.out.print("Введите автора: ");
        String authorName = scanner.nextLine();
        System.out.println("Книги " + authorName + ":");
        for (Book book : books) {
            if (book.getAuthor().equalsIgnoreCase(authorName)) {
                book.displayInfo();
            }
        }

        // b) список книг, выпущенных заданным издательством
        System.out.print("Введите название издательства: ");
    }

```

```

        String publisherName = scanner.nextLine();
        System.out.println("Книги " + publisherName + ":");
        for (Book book : books) {
            if
(book.getPublisher().equalsIgnoreCase(publisherName)) {
                book.displayInfo();
            }
        }

        // с) список книг, выпущенных после заданного года
        System.out.print("Введите год издательства: ");
        int year = scanner.nextInt();
        System.out.println("Книги изданные после " + year +
":");
        for (Book book : books) {
            if (book.getYear() > year) {
                book.displayInfo();
            }
        }
    }
}

```

Работа программы показана на рисунке 4.

```

Введите автора: Достоевский
Книги Достоевский:
Название: Преступление и наказание
Автор(ы): Достоевский
Издательство: Москва
Год: 2000
Количество страниц: 300
Цена: 20.0 руб.
Переплет: Твердый

Введите название издательства: Аст
Книги Аст:
Название: 1984
Автор(ы): Оруэлл
Издательство: АСТ
Год: 2010
Количество страниц: 400
Цена: 30.0 руб.
Переплет: Твердый

Введите год издательства: 2010
Книги изданные после 2010:

```

Рисунок 4 – Работа программы

**Задание 5:** создать приложение, удовлетворяющее требованиям, приведенным в задании. Аргументировать принадлежность классу каждого

создаваемого метода и корректно переопределить для каждого класса методы equals(), hashCode(), toString(). Создать объект класса Простая дробь, используя класс Число. Методы: вывод на экран, сложение, вычитание, умножение, деление.

#### Код класса Main:

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        // Ввод первой дроби
        System.out.println("Введите первую дробь (числитель и знаменатель):");
        int numerator1 = scanner.nextInt();
        int denominator1 = scanner.nextInt();
        SimpleFraction fraction1 = new SimpleFraction(numerator1, denominator1);

        // Ввод второй дроби
        System.out.println("Введите вторую дробь (числитель и знаменатель):");
        int numerator2 = scanner.nextInt();
        int denominator2 = scanner.nextInt();
        SimpleFraction fraction2 = new SimpleFraction(numerator2, denominator2);

        SimpleFraction sum = fraction1.addition(fraction2);
        System.out.println("Сумма простых дробей: " + sum);

        SimpleFraction difference = fraction1.subtraction(fraction2);
        System.out.println("Разность простых дробей: " + difference);

        SimpleFraction product = fraction1.multiplication(fraction2);
        System.out.println("Произведение: " + product);

        SimpleFraction quotient = fraction1.division(fraction2);
        System.out.println("Частное: " + quotient);

        // Ввод первого числа
        System.out.println("Введите первое число:");
        int number1Value = scanner.nextInt();
        Number number1 = new Number(number1Value);

        // Ввод второго числа
        System.out.println("Введите второе число:");
        int number2Value = scanner.nextInt();
        Number number2 = new Number(number2Value);
    }
}
```

```

        Number sumNum = number1.addition(number2);
        System.out.println("Сумма: " + sumNum);

        Number differenceNum = number1.subtraction(number2);
        System.out.println("Разность: " + differenceNum);

        Number productNum = number1.multiplication(number2);
        System.out.println("Произведение: " + productNum);

        Number quotientNum = number1.division(number2);
        System.out.println("Частное: " + quotientNum);
    }
}

```

### Код класса SimpleFraction:

```

import java.util.Objects;
public class SimpleFraction {
    private int numerator;
    private int denominator;

    public SimpleFraction(int numerator, int denominator) {
        this.numerator = numerator;
        this.denominator = denominator;
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj) {
            return true;
        }
        if (obj == null || getClass() != obj.getClass()) {
            return false;
        }
        SimpleFraction otherFraction = (SimpleFraction) obj;
        return numerator == otherFraction.numerator &&
            denominator == otherFraction.denominator;
    }

    @Override
    public int hashCode() {
        return Objects.hash(numerator, denominator);
    }

    @Override
    public String toString() {
        return numerator + "/" + denominator;
    }

    public SimpleFraction addition(SimpleFraction otherFraction)
{

```

```

        int      newNumerator      =      this.numerator      *
otherFraction.denominator      +      otherFraction.numerator      *
this.denominator;
        int      newDenominator      =      this.denominator      *
otherFraction.denominator;
        return new SimpleFraction(newNumerator, newDenominator);
    }

    public      SimpleFraction      subtraction(SimpleFraction
otherFraction) {
        int      newNumerator      =      this.numerator      *
otherFraction.denominator      -      otherFraction.numerator      *
this.denominator;
        int      newDenominator      =      this.denominator      *
otherFraction.denominator;
        return new SimpleFraction(newNumerator, newDenominator);
    }

    public      SimpleFraction      multiplication(SimpleFraction
otherFraction) {
        int      newNumerator      =      this.numerator      *
otherFraction.numerator;
        int      newDenominator      =      this.denominator      *
otherFraction.denominator;
        return new SimpleFraction(newNumerator, newDenominator);
    }

    public SimpleFraction division(SimpleFraction otherFraction)
{
        int      newNumerator      =      this.numerator      *
otherFraction.denominator;
        int      newDenominator      =      this.denominator      *
otherFraction.numerator;
        return new SimpleFraction(newNumerator, newDenominator);
    }
}

```

### **Код класса Number:**

```

import java.util.Objects;
public class Number {
    private int value;

    public Number(int value) {
        this.value = value;
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj) {
            return true;
        }
        if (obj == null || getClass() != obj.getClass()) {
            return false;
        }
    }
}

```

```

        }
        Number otherNumber = (Number) obj;
        return value == otherNumber.value;
    }

    @Override
    public int hashCode() {
        return Objects.hash(value);
    }

    @Override
    public String toString() {
        return String.valueOf(value);
    }

    public Number addition(Number otherNumber) {
        int sum = this.value + otherNumber.value;
        return new Number(sum);
    }

    public Number subtraction(Number otherNumber) {
        int difference = this.value - otherNumber.value;
        return new Number(difference);
    }

    public Number multiplication(Number otherNumber) {
        int product = this.value * otherNumber.value;
        return new Number(product);
    }

    public Number division(Number otherNumber) {
        if (otherNumber.value == 0) {
            throw new ArithmeticException("Division by zero");
        }
        int quotient = this.value / otherNumber.value;
        return new Number(quotient);
    }
}

```

**Работа программы показана на рисунке 5.**



```
Введите первую дробь (числитель и знаменатель):  
1  
2  
Введите вторую дробь (числитель и знаменатель):  
3  
4  
Сумма простых дробей: 10/8  
Разность простых дробей: -2/8  
Произведение: 3/8  
Частное: 4/6  
Введите первое число:  
2  
Введите второе число:  
3  
Сумма: 5  
Разность: -1  
Произведение: 6  
Частное: 0
```

Рисунок 5 – Работа программы

**Задание 6:** создать приложение, удовлетворяющее требованиям, приведенным в задании. Аргументировать принадлежность классу каждого создаваемого метода и корректно переопределить для каждого класса методы `equals()`, `hashCode()`, `toString()`. Создать объект класса Дом, используя классы Окно, Дверь. Методы: закрыть на ключ, вывести на консоль количество окон, дверей.

Код класса Main:

```
public class Main {  
    public static void main(String[] args) {  
        Window window1 = new Window(false);  
        Window window2 = new Window(true);  
  
        Door door1 = new Door(false);  
        Door door2 = new Door(true);  
  
        Window[] windows = {window1, window2};  
        Door[] doors = {door1, door2};  
  
        House house = new House(windows, doors);  
        house.printWindowsAndDoorsCount();  
  
        house.lockOnKey();  
        house.closeWindow(0);  
        house.unlockDoor(1);  
  
        house.printWindowsAndDoorsCount();  
    }  
}
```

### Код класса House:

```
class House {
    private Window[] windows;
    private Door[] doors;

    public House(Window[] windows, Door[] doors) {
        this.windows = windows;
        this.doors = doors;
    }

    public void lockOnKey() {
        for (Door door : doors) {
            door.lock();
        }
    }

    public void printWindowsAndDoorsCount() {
        System.out.println("Количество окон: " +
windows.length);
        System.out.println("Количество дверей: " +
doors.length);
    }

    public void openWindow(int index) {
        if (index >= 0 && index < windows.length) {
            windows[index].open();
            System.out.println("Окно номер " + index + "
открыто.");
        } else {
            System.out.println("Столько окон нет");
        }
    }

    public void closeWindow(int index) {
        if (index >= 0 && index < windows.length) {
            windows[index].close();
            System.out.println("Окно номер " + index + "
закрыто.");
        } else {
            System.out.println("Столько окон нет");
        }
    }

    public void unlockDoor(int index) {
        if (index >= 0 && index < doors.length) {
            doors[index].unlock();
            System.out.println("Дверь номер " + index + "
открыта.");
        } else {
            System.out.println("Столько дверей нет");
        }
    }
}
```

```

        public void lockDoor(int index) {
            if (index >= 0 && index < doors.length) {
                doors[index].lock();
                System.out.println("Дверь номер " + index + "
закрыта.");
            } else {
                System.out.println("Столько дверей нет");
            }
        }
    }
}

```

### Код класса Door:

```

import java.util.Objects;
class Door {
    private boolean isLocked;

    public Door(boolean isLocked) {
        this.isLocked = isLocked;
    }

    public void lock() {
        isLocked = true;
    }

    public void unlock() {
        isLocked = false;
    }

    @Override
    public String toString() {
        return "Door {isLocked=" + isLocked + "}";
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj) {
            return true;
        }
        if (obj == null || getClass() != obj.getClass()) {
            return false;
        }
        Door door = (Door) obj;
        return isLocked == door.isLocked;
    }

    @Override
    public int hashCode() {
        return Objects.hash(isLocked);
    }
}

```

### Код класса Window:

```
import java.util.Objects;
class Window {
    private boolean isOpen;

    public Window(boolean isOpen) {
        this.isOpen = isOpen;
    }

    public void open() {
        isOpen = true;
    }

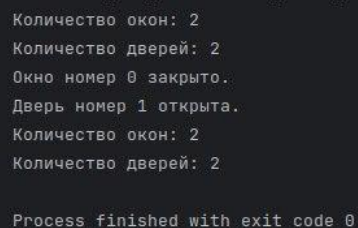
    public void close() {
        isOpen = false;
    }

    @Override
    public String toString() {
        return "Window {isOpen=" + isOpen + "}";
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj) {
            return true;
        }
        if (obj == null || getClass() != obj.getClass()) {
            return false;
        }
        Window window = (Window) obj;
        return isOpen == window.isOpen;
    }

    @Override
    public int hashCode() {
        return Objects.hash(isOpen);
    }
}
```

Работа программы показана на рисунке 6.



```
Количество окон: 2
Количество дверей: 2
Окно номер 0 закрыто.
Дверь номер 1 открыта.
Количество окон: 2
Количество дверей: 2

Process finished with exit code 0
```

Рисунок 6 – Работа программы

**Задание 7:** построить модель программной системы. Система Вступительные экзамены. Абитуриент регистрируется на Факультет, сдает Экзамены. Преподаватель выставяет Оценку. Система подсчитывает средний балл и определяет Абитуриентов, зачисленных в учебное заведение.

**Код класса Main:**

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Введите проходной балл на факультет:
");
        int passingScore = scanner.nextInt();
        scanner.nextLine();

        Faculty faculty = new Faculty(passingScore);

        System.out.print("Введите количество абитуриентов: ");
        int numApplicants = scanner.nextInt();
        scanner.nextLine();

        for (int i = 0; i < numApplicants; i++) {
            System.out.print("Введите ФИО абитуриента: ");
            String name = scanner.nextLine();
            Abiturient abiturient = new Abiturient(name);

            System.out.print("Введите количество экзаменов: ");
            int numScores = scanner.nextInt();
            scanner.nextLine();

            for (int j = 0; j < numScores; j++) {
                System.out.print("Введите баллы за экзамен: ");
                int score = scanner.nextInt();
                scanner.nextLine();
                abiturient.addExamScore(score);
            }

            faculty.addApplicant(abiturient);
        }

        System.out.println("Результаты поступления:");
        faculty.enrollStudents();
    }
}
```

**Код класса Faculty:**

```
import java.util.ArrayList;
import java.util.List;
```

```

class Faculty {
    private List<Abiturient> abiturients;
    private int passingScore;

    public Faculty(int passingScore) {
        this.abiturients = new ArrayList<>();
        this.passingScore = passingScore;
    }

    public void addApplicant(Abiturient abiturient) {
        abiturients.add(abiturient);
    }

    public void enrollStudents() {
        for (Abiturient abiturient : abiturients) {
            if (abiturient.calculateAverageScore() >=
passingScore) {
                System.out.println("Абитуриент " +
abiturient.getName() + " принят на факультет.");
            } else {
                System.out.println("Абитуриент " +
abiturient.getName() + " не принят на факультет.");
            }
        }
    }
}

```

#### **Код класса Abiturient:**

```

import java.util.ArrayList;
import java.util.List;

class Abiturient {
    private String name;
    private List<Integer> examScores;

    public Abiturient(String name) {
        this.name = name;
        this.examScores = new ArrayList<>();
    }

    public void addExamScore(int score) {
        examScores.add(score);
    }

    public int calculateAverageScore() {
        int sum = 0;
        for (int score : examScores) {
            sum += score;
        }
        return examScores.size() > 0 ? sum /
examScores.size() : 0;
    }

    public String getName() {

```

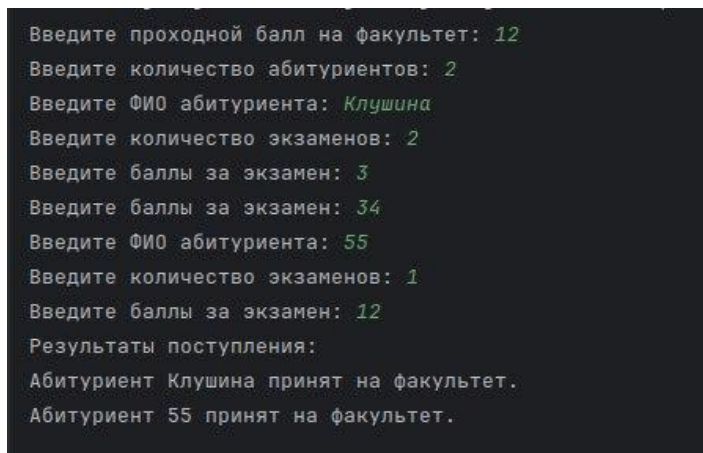
```

        return name;
    }

    @Override
    public String toString() {
        return "Абитуриент{" +
            "имя: " + name + '\'' +
            ", , баллы за экзамены: " + examScores +
            '\'';
    }
}

```

Работа программы показана на рисунке 7.



```

Введите проходной балл на факультет: 12
Введите количество абитуриентов: 2
Введите ФИО абитуриента: Клушина
Введите количество экзаменов: 2
Введите баллы за экзамен: 3
Введите баллы за экзамен: 34
Введите ФИО абитуриента: 55
Введите количество экзаменов: 1
Введите баллы за экзамен: 12
Результаты поступления:
Абитуриент Клушина принят на факультет.
Абитуриент 55 принят на факультет.

```

Рисунок 7 – Работа программы

**Задание 8:** построить модель программной системы. Система Вступительные экзамены. Абитуриент регистрируется на Факультет, сдает Экзамены. Преподаватель выставяет Оценку. Система подсчитывает средний балл и определяет Абитуриентов, зачисленных в учебное заведение.

Код класса Main:

```

import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Library library = new Library();

        Librarian librarian = new Librarian("Ксения");
        Reader reader = new Reader("Настя");

        // Взаимодействие с библиотекой
        System.out.print("Введите название книги: ");
        String bookTitle = scanner.nextLine();

        Book foundBook = library.findBook(bookTitle);
    }
}

```

```

        if (foundBook != null) {
            librarian.issueBook(foundBook, reader);
        } else {
            System.out.println("Книга не найдена в каталоге.");
        }
        Book newBook = new Book("Преступление и наказание",
"Достоевский");
        library.addBook(newBook);

        // Взаимодействие с библиотекой
        System.out.print("Введите название книги: ");
        String bookTitle2 = scanner.nextLine();

        Book foundBook2 = library.findBook(bookTitle2);

        if (foundBook2 != null) {
            librarian.issueBook(foundBook2, reader);
        } else {
            System.out.println("Книга не найдена в каталоге.");
        }

        System.out.print("Читатель вернул книгу? (да/нет): ");
        String returnStatus = scanner.nextLine();

        if (returnStatus.equalsIgnoreCase("нет")) {
            reader.addToBlacklist();
        }

        scanner.close();
    }
}

```

### **Код класса Book:**

```

class Book {
    private String title;
    private String author;
    private boolean isAvailable;

    public Book(String title, String author) {
        this.title = title;
        this.author = author;
        this.isAvailable = true;
    }

    public String getTitle() {
        return title;
    }

    public String getAuthor() {
        return author;
    }

    public boolean isAvailable() {

```



```

        return isAvailable;
    }

    public void setAvailable(boolean available) {
        isAvailable = available;
    }

    @Override
    public String toString() {
        return title + " от " + author;
    }
}

```

#### **Код класса Librarian:**

```

class Librarian {
    private String name;

    public Librarian(String name) {
        this.name = name;
    }

    public void issueBook(Book book, Reader reader) {
        if (book.isAvailable() && !reader.isOnBlacklist()) {
            book.setAvailable(false);
            System.out.println(this.name + " выдал книгу '" +
book.getTitle() + "' " + reader.getName());
        } else {
            System.out.println("НЕЛЬЗЯ выдать книгу " +
reader.getName());
        }
    }
}

```

#### **Код класса Library:**

```

import java.util.ArrayList;
import java.util.List;
class Library {
    private List<Book> catalog;
    private List<String> blacklist;

    public Library() {
        this.catalog = new ArrayList<>();
        this.blacklist = new ArrayList<>();
    }

    public void addBook(Book book) {
        catalog.add(book);
    }

    public boolean checkBlacklist(String readerName) {
        return blacklist.contains(readerName);
    }
}

```

```

        public Book findBook(String title) {
            for (Book book : catalog) {
                if (book.getTitle().equalsIgnoreCase(title)    &&
book.isAvailable()) {
                    return book;
                }
            }
            return null;
        }

        public void lendBook(Book book, String readerName) {
            book.setAvailable(false);
            System.out.println(readerName + "   вернул   книгу:  " +
book);
        }

        public void addReaderToBlacklist(String readerName) {
            blacklist.add(readerName);
            System.out.println(readerName + "   добавлен   в   черный
список.");
        }
    }
}

```

#### **Код класса Reader:**

```

class Reader {
    private String name;
    private boolean onBlacklist;

    public Reader(String name) {
        this.name = name;
        this.onBlacklist = false;
    }

    public String getName() {
        return name;
    }

    public boolean isOnBlacklist() {
        return onBlacklist;
    }

    public void addToBlacklist() {
        this.onBlacklist = true;
        System.out.println(this.name + "   добавлена   в   черный
список.");
    }
}

```

**Работа программы показана на рисунке 8.**

```
Книга не найдена в каталоге.  
Введите название книги: "Преступление и наказание"  
Книга не найдена в каталоге.  
Читатель вернул книгу? (да/нет): да
```

Рисунок 8 – Работа программы

**Вывод:** были освоены принципы ООП на языке программирования  
Java