

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”
**КАФЕДРА ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ**

ОТЧЁТ
по лабораторной работе №7

Выполнила
студентка группы ПО-9
Тупик Ю. Л.

Проверил:
Крощенко А. А.

Брест 2024

Цель работы: освоить возможности языка программирования Java в построении графических приложений.

Задание 1

8) Задать движение по экрану строк (одна за другой) из массива строк. Направление движения по апплету и значение каждой строки выбираются случайным образом.

Выполнение задания

Код программы

```
import javax.swing.*;
import java.awt.*;
import java.util.ArrayList;
import java.util.Random;
import java.util.Scanner;

public class Task1 extends JPanel implements Runnable {
    private String[] strings; // Массив строк для отображения
    private int[] xPos; // Массив для хранения текущих горизонтальных
    позиций строк
    private int[] yPos; // Массив для хранения текущих вертикальных позиций
    строк
    private int[] xVel; // Массив для хранения скоростей перемещения по
    горизонтали
    private int[] yVel; // Массив для хранения скоростей перемещения по
    вертикали
    private int numStrings; // Количество строк
    private int speed; // Скорость движения строк

    // Конструктор класса
    public Task1(String[] strings, int speed) {
        this.strings = strings;
        this.speed = speed;
        numStrings = strings.length;
        xPos = new int[numStrings];
        yPos = new int[numStrings];
        xVel = new int[numStrings];
        yVel = new int[numStrings];
    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        // Отображение каждой строки на экране
        for (int i = 0; i < numStrings; i++) {
            g.drawString(strings[i], xPos[i], yPos[i]);
        }
    }

    @Override
    public void run() {
        Random random = new Random();
        // Инициализация начальных позиций и скоростей строк
```

```

for (int i = 0; i < numStrings; i++) {
    xPos[i] = random.nextInt(getWidth());
    yPos[i] = random.nextInt(getHeight());

    xVel[i] = random.nextInt(numStrings) - (numStrings / 2);
    yVel[i] = random.nextInt(numStrings) - (numStrings / 2);
}

while (true) {
    moveStrings(); // Перемещение строк
    repaint(); // Перерисовка экрана
    try {
        Thread.sleep(this.speed); // Задержка для скорости движения
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}

// Метод для перемещения строк
private void moveStrings() {
    for (int i = 0; i < numStrings; i++) {
        xPos[i] += xVel[i];
        yPos[i] += yVel[i];

        // Проверка на выход строки за границы экрана, если это
        // происходит, меняем направление скорости
        if (xPos[i] < 0 || xPos[i] > getWidth()) {
            xVel[i] = -xVel[i];
        }

        if (yPos[i] < 0 || yPos[i] > getHeight()) {
            yVel[i] = -yVel[i];
        }
    }
}

public static void main(String[] args) {
    ArrayList<String> strings = new ArrayList<>();

    Scanner scanner = new Scanner(System.in);

    System.out.println("Введите строки (для завершения введите пустую
строку):");

    String input;
    while (!(input = scanner.nextLine()).isEmpty()) {
        strings.add(input);
    }

    System.out.println("Скорость движения строк: ");
    int speed = scanner.nextInt();

    // Создание графического окна и запуск анимации

```

```

        JFrame frame = new JFrame("Moving Strings");
        Task1 movingStrings = new Task1(strings.toArray(new String[0]),
speed);
        frame.add(movingStrings);
        frame.setSize(1000, 700); // Размер окна
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);

        Thread thread = new Thread(movingStrings);
        thread.start();
    }
}

```

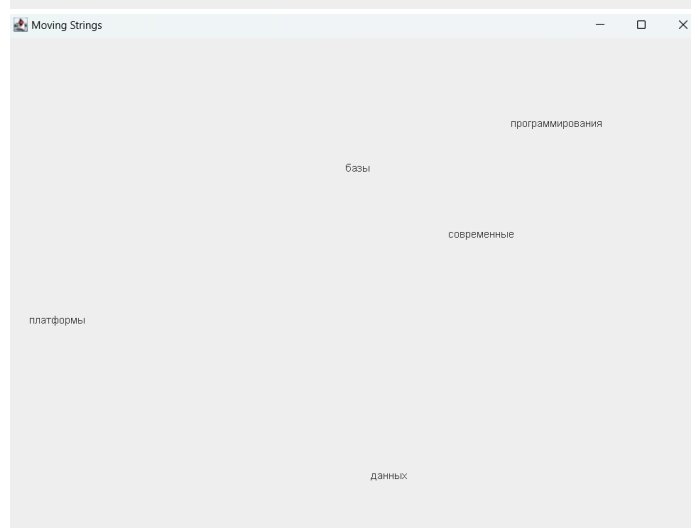
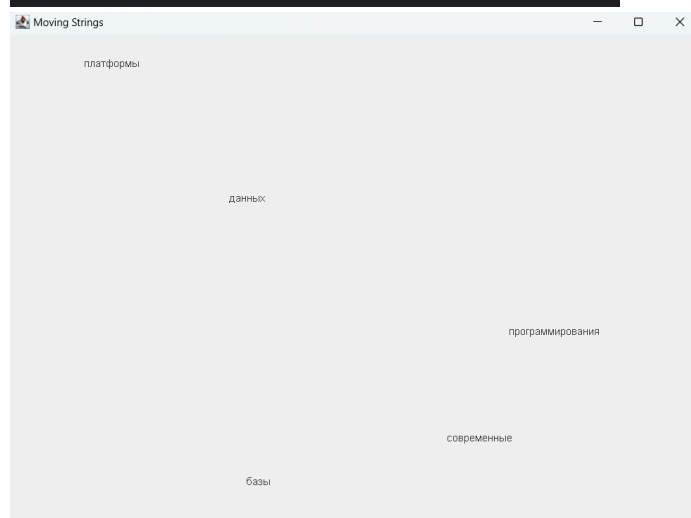
Результат

```

C:\Users\Юлия\.jdk\corretto-17.0.9\bin\java.exe "-javaag
Введите строки (для завершения введите пустую строку):
базы
данных
современные
платформы
программирования

Скорость движения строк:
20

```



Задание 2

Реализовать построение заданного типа фрактала

8) Кривая дракона

Выполнение задания

Код программы

```
import javax.swing.*;
import java.awt.*;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import java.util.Scanner;

public class Task2 extends JFrame {
    private List<Integer> turns; // Список поворотов для построения
    фрактала
    private double startingAngle, sideLength; // Начальный угол и длина
    стороны для отрисовки
    private static final double ANGLE_45DEGREES = Math.PI / 4; // Угол в
    радианах для поворота на 45 градусов

    // Конструктор класса, принимает количество итераций для построения
    фрактала
    public Task2(int amountOfIterations) {
        super("Dragon Curve");
        setBounds(100, 100, 1000, 800); // Установка размеров и положения
        окна
        setDefaultCloseOperation(EXIT_ON_CLOSE); // Установка операции по
        закрытию окна
        turns = getSequence(amountOfIterations); // Получение
        последовательности поворотов
        startingAngle = -amountOfIterations * ANGLE_45DEGREES; // Начальный
        угол для отрисовки
        sideLength = 550 / Math.pow(2, amountOfIterations / 2.); // Длина
        стороны фрактала
    }

    // Метод для получения последовательности поворотов
    public List<Integer> getSequence(int iterations) {
        List<Integer> turnSequence = new ArrayList<>(); // Инициализация
        списка поворотов
        for (int i = 0; i < iterations; i++) { // Цикл по количеству
        итераций
            List<Integer> copy = new ArrayList<>(turnSequence); // Создание
            копии текущей последовательности
            Collections.reverse(copy); // Реверс списка
            turnSequence.add(1); // Добавление поворота вправо
            for (Integer turn : copy) { // Цикл по копии последовательности
                turnSequence.add(-turn); // Добавление поворота влево,
            инвертируя повороты из копии
            }
        }
    }
}
```

```

        return turnSequence; // Возврат полученной последовательности
поворотов
    }

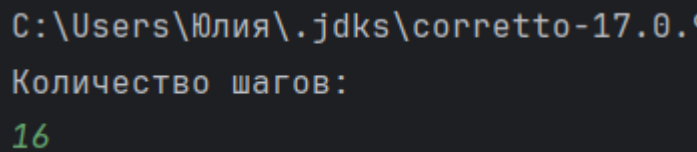
    // Переопределение метода отрисовки компонента
    @Override
    public void paint(Graphics g) {
        g.setColor(Color.orange); // Установка цвета для рисования
        double angle = startingAngle; // Установка начального угла
        int x1 = 230, y1 = 450; // Начальные координаты
        int x2 = x1 + (int) (Math.cos(angle) * sideLength); // Вычисление
конечных координат
        int y2 = y1 + (int) (Math.sin(angle) * sideLength);
        g.drawLine(x1, y1, x2, y2); // Отрисовка начальной линии
        x1 = x2; // Обновление начальных координат
        y1 = y2;
        for (Integer turn : turns) { // Цикл по последовательности
поворотов
            angle += turn * (Math.PI / 2); // Вычисление угла поворота
            x2 = x1 + (int) (Math.cos(angle) * sideLength); // Вычисление
конечных координат
            y2 = y1 + (int) (Math.sin(angle) * sideLength);
            g.drawLine(x1, y1, x2, y2); // Отрисовка линии
            x1 = x2; // Обновление начальных координат
            y1 = y2;
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Количество шагов: ");
        int amountOfIterations = scanner.nextInt();

        new Task2(amountOfIterations).setVisible(true);
    }
}

```

Результат



C:\Users\Юлия\.jdk\corretto-17.0.6
 Количество шагов:
 16



Вывод: освоила возможности языка программирования Java в построении графических приложений.