

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Факультет прикладной математики и информатики

Кафедра информационных систем управления

Отчёт

по учебной практике (2 курс)

Выполнил студент 12 группы
Шишлянников Иван Викторович

Минск 2020

Оглавление

ЗАДАНИЕ №1	4
УСЛОВИЕ ЗАДАЧИ	4
ОСОБЕННОСТИ РЕАЛИЗАЦИИ	4
ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ	6
ЗАДАНИЕ №2	7
УСЛОВИЕ ЗАДАЧИ	7
ОСОБЕННОСТИ РЕАЛИЗАЦИИ	7
ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ	8
ЗАДАНИЕ №3	9
УСЛОВИЕ ЗАДАЧИ	9
ОСОБЕННОСТИ РЕАЛИЗАЦИИ	9
ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ	12
ЗАДАНИЕ №4	13
УСЛОВИЕ ЗАДАЧИ	13
ОСОБЕННОСТИ РЕАЛИЗАЦИИ	13
ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ	15
ЗАДАНИЕ №5	17
УСЛОВИЕ ЗАДАЧИ	17
ОСОБЕННОСТИ РЕАЛИЗАЦИИ	17
ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ	19
ЗАДАНИЕ №6	20
УСЛОВИЕ ЗАДАЧИ	20
ОСОБЕННОСТИ РЕАЛИЗАЦИИ	20
ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ	22
ЗАДАНИЕ №7	24
УСЛОВИЕ ЗАДАЧИ	24
ОСОБЕННОСТИ РЕАЛИЗАЦИИ	24
ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ	25
ЗАДАНИЕ №8	25
УСЛОВИЕ ЗАДАЧИ	25

ОСОБЕННОСТИ РЕАЛИЗАЦИИ	25
ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ	26
ЗАДАНИЕ №9	27
УСЛОВИЕ ЗАДАЧИ	27
ОСОБЕННОСТИ РЕАЛИЗАЦИИ	27
ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ	28
ЗАДАНИЕ №10	29
УСЛОВИЕ ЗАДАЧИ	29
ОСОБЕННОСТИ РЕАЛИЗАЦИИ	29
ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ	29
ЗАДАНИЕ №11	31
УСЛОВИЕ ЗАДАЧИ	31
ОСОБЕННОСТИ РЕАЛИЗАЦИИ	31
ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ	31
ЗАДАНИЕ №12	34
УСЛОВИЕ ЗАДАЧИ	34
ОСОБЕННОСТИ РЕАЛИЗАЦИИ	34
ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ	35

Задание №1

Условие задачи

Изобразить прямоугольник и вписанный в него эллипс вращающимися в плоскости экрана вокруг своего центра в противоположных направлениях:

- для изображения указанной в задании фигуры создать класс, реализующий интерфейс

Shape (можно взять базовым библиотечный класс, реализующий Shape)

- выполнить указанные в задании перемещения указанной фигуры с помощью аффинного

преобразования координат

- выполнить рисунок в окне фрейма с выбранной толщиной границы фигуры, цветом

границы и цветом внутренней области (вводить толщину и цвет в качестве аргументов

ваших программ).

Особенности реализации

Обновление аффинных преобразований рекуррентно исходя из предыдущих при каждой перерисовке:

```

public void paint(Graphics g) {

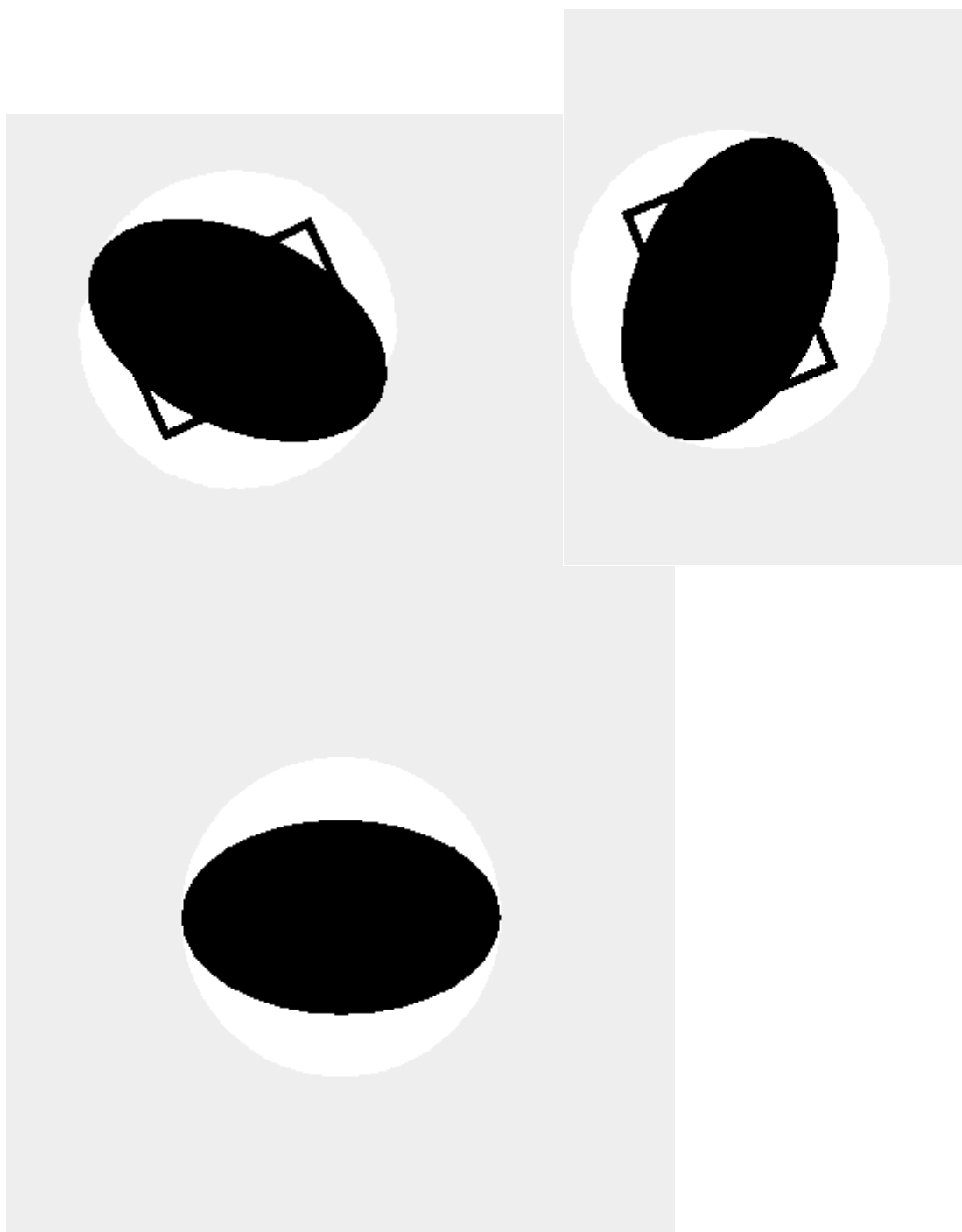
    Graphics2D graphics2D = (Graphics2D)g;
    graphics2D.setStroke(new BasicStroke(LineWidth));
    graphics2D.setColor(Color.WHITE);
    graphics2D.draw(shapeToDraw);
    graphics2D.draw(r1);
    graphics2D.fill(r1);
    graphics2D.setColor(FigurebrushColor);
    r1=ellipseAffinetransform.createTransformedShape(r1);
    shapeToDraw = affineTransform.createTransformedShape(shapeToDraw);
    graphics2D.draw(r1);
    graphics2D.setColor(EntireColor);
    graphics2D.fill(r1);
    graphics2D.setColor(FigurebrushColor);
    graphics2D.draw(shapeToDraw);

    System.out.println("Repainting "+shapeToDraw);
}

```

Перерисовка происходит каждые 120 ms в отдельном потоке.

Демонстрация работы программы



Задание №2

Условие задачи

Создайте тестовое приложение (Frame/JFrame) для демонстрации вашего решения, при этом: - для изображения указанной в задании фигуры создать класс, реализующий интерфейс Shape; - создайте указанный фильтр изображения; при тестировании выведите фигуру без фильтра и с фильтром (аналогично фильтрам из примеров); - моделируйте освещение и тень от объекта при помощи альфа-канала и/или механизма обработки изображения; - при рисовании используйте сглаживание, внеэкранный буфер и преобразования координат.

Фигура (дорожный знак): надпись 80 в окружности, цвет окружности и символа – красный, цвет фона – серый с градиентной заливкой справа-налево

Фильтр: Brighten

Особенности реализации

Рисование изображения

```
public static void DrawShape(Graphics2D graphics2D, Shape MyCircle, Font MyFont){
    graphics2D.setStroke(new BasicStroke( width: 30));

    graphics2D.setColor(Color.WHITE);
    graphics2D.fillRect( i: 0, i1: 0, i2: 700, i3: 350);
    Rectangle2D bounds = MyCircle.getBounds2D();
    GradientPaint gradientPaint = new GradientPaint((float)bounds.getMaxX(), (float)bounds.getY(), Color.GRAY,
        (float)bounds.getY(), (float)bounds.getX(), Color.white);
    graphics2D.setPaint(gradientPaint);
    graphics2D.fill(MyCircle);

    graphics2D.setColor(Color.red);
    graphics2D.draw(MyCircle);

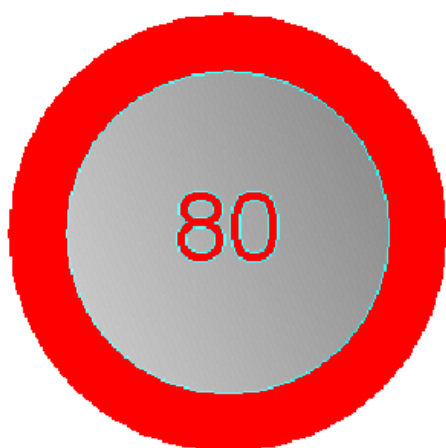
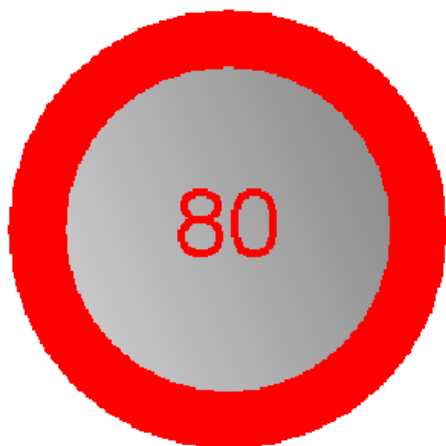
    graphics2D.setFont(MyFont);

    float textWidth= graphics2D.getFontMetrics().stringWidth("80");
    graphics2D.drawString( s: "80", v: (float)MyCircle.getBounds().getCenterX()-textWidth/2, v1: (float) MyCircle.getBounds().getCenterY()+15);
}
```

Накладывание фильтра на изображение.

```
public BufferedImage processImage(BufferedImage image) {
    float[] brightonMatrix = { 0, -1.0f, 0.0f, -1.0f, 5.0f, -1.0f, 0.0f, -1.0f, 0.0f };
    BufferedImageOp brightenFilter = new ConvolveOp(new Kernel( width: 3, height: 3, brightonMatrix:
        ConvolveOp.EDGE_NO_OP, hints: null));
    return brightenFilter.filter(image, bufferedImage1: null);
}
```

Демонстрация работы программы



Задание №3

Условие задачи

- 1) Разработайте пользовательский класс Shape реализующий рисование указанной алгебраической линии.
- 2) Разработайте пользовательский класс Stroke для отображения указанного контура, используя в качестве исходных точек результаты класса Shape, созданного на шаге 1)
- 3) Создайте приложение (Frame / JFrame) для тестирования и демонстрации разработанных классов.

9. Линия: Астроида

$$(x^2 + y^2 - a^2)^3 + 27x^2y^2a^2 = 0$$

Контур: 

Особенности реализации

Чтобы удобно рисовать фигуру будем работать с параметрическим видом данной функции.

```
public int GetX(float t){  
    return centerX+ (int) Math.round(((200*Math.pow(Math.sin(t),3))));  
}  
  
public int GetY(float t){  
    return centerY+ (int) Math.round((200*Math.pow(Math.cos(t),3)));  
}
```

Рисование каждой точки будем делать с определенным шагом.

```
point2DS.clear();
for (float t=Tmin;t<Tmax;t+=jump){
    int x = GetX(t);
    int y = GetY(t);
    g.fillRect(x,y, 2, 2);
    point2DS.add(new Point(x,y));
}

point2DS.sort(comparator);

int yMin = point2DS.get(0).y;

result=new ArrayList<Point>();

for(int i=0;i<point2DS.size() ;i++){
    if(point2DS.get(i).y<=yMin) {
        result.add(point2DS.get(i));
    }
}
point2DS.clear();
int prevSize =result.size();
for(int i =prevSize-1;i>=0;i--){

    result.add(new Point(result.get(i).x, y: 2*yMin- result.get(i).y));
}
```

Отрисовываем контур в своем классе MyStroke

```
ArrayList<Point> points = s.GetPoints();

for (int i=0;i<points.size();i+=3){

    if(i+5>=points.size()){
        break;
    }
    int vX = points.get(i+2).x-points.get(i).x;
    int vY = points.get(i+2).y-points.get(i).y;
    int nvX =-vY;
    int nvY =vX;

    nvY=-vX;
    nvX=vY;

    int vLength =(int) Math.sqrt( nvX*nvX+nvY*nvY);

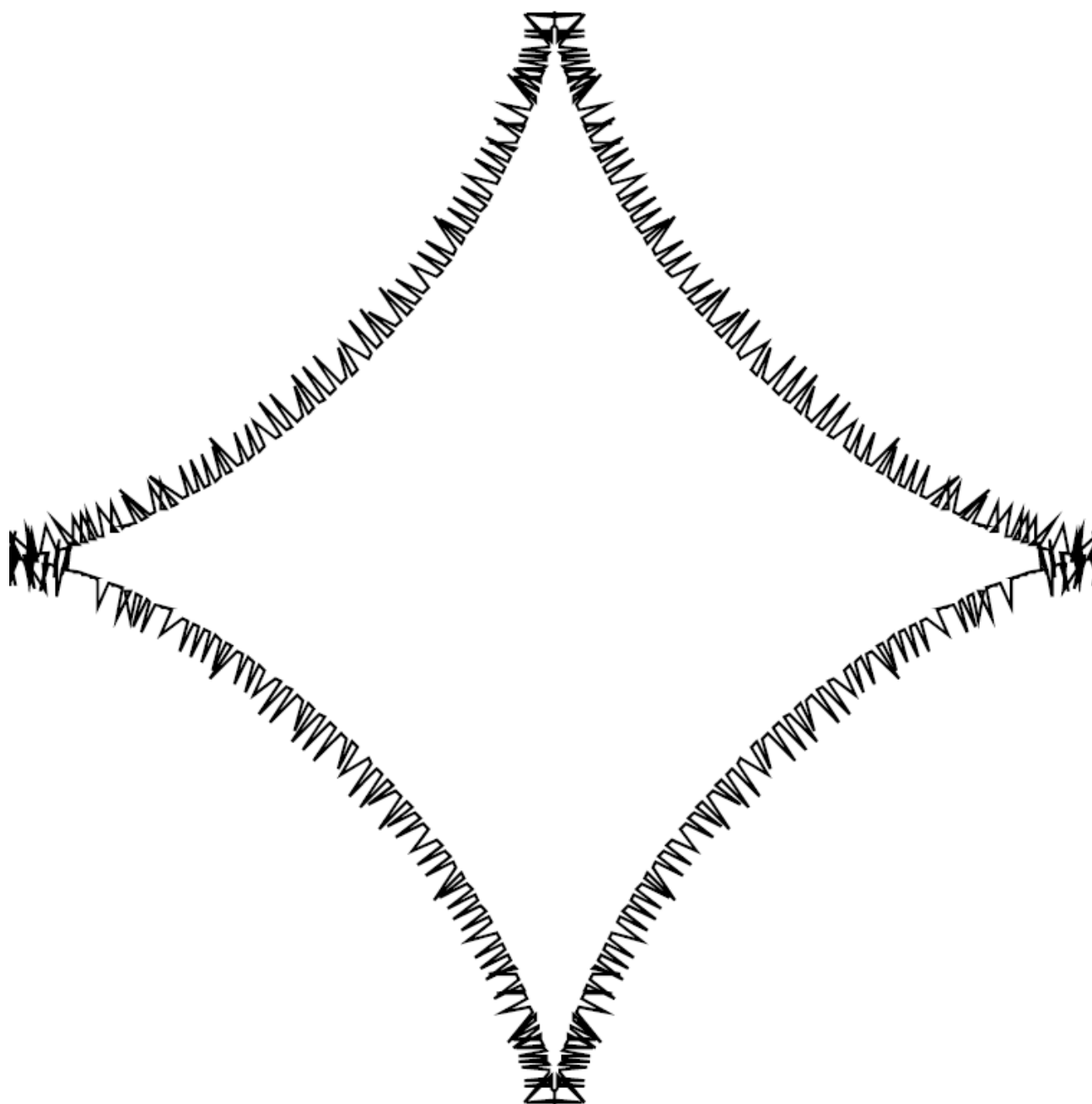
    if (vLength==0){
        vLength = 1;
    }

    points.get(i+1).x+=10*nvX/vLength;
    points.get(i+1).y+=10*nvY/vLength;
}

Polygon myPolygon = new Polygon();
for (int i=0;i<points.size();i++){

    myPolygon.addPoint(points.get(i).x,points.get(i).y);
}
```

Демонстрация работы программы



Задание №4

Условие задачи

Для выполнения задания используется ваш вариант решения задания №3. Модифицируйте вашу программу следующим образом. В демонстрационное приложение добавьте возможность печати небольшого отчёта о решении задания №3. Отчёт должен содержать следующее:

- рисунок с подписью (!) алгебраической линии вашего задания
- исходный текст класса Shape, реализующий рисование указанной алгебраической линии;

для длинных строк, выходящих за границы области печати, организуйте перенос текста на новую строку с разрывом по пробельным символам.

При печати используйте режим альбомной ориентации страницы и двустороннюю печать.

Рисунок должен занимать не более половины страницы, при печати выравнивать его по горизонтали.

Особенности реализации

Для печати был создан класс PlotFrame, который реализует интерфейс Printable

Получаем текст кода:

```
private PlotFrame() {
    initComponents();
    this.setTitle("04 - astroid");
    this.setSize( width: 400, height: 470);
    File f = new File( pathname: "src");
    if (f.exists()) {
        System.out.println(f.getAbsolutePath());
    }
    textlines = initTextLines(new File( pathname: "C:/BSU-Season-4/UP/Lab4/src/com/company/AstroIdShape.java"));
}
```

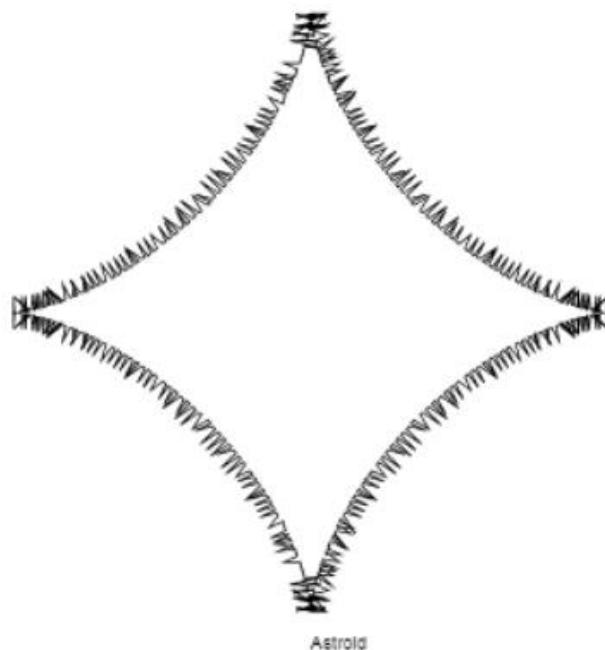
Рассчитываем разметку страницу по количеству строк и их размера.

```
if (pageBreaks == null) {
    System.out.println(textLines.length);
    int linesPerPage = (int) (pageFormat.getImageableHeight() / lineHeight);
    int numBreaks = (textLines.length - 1) / linesPerPage + 1;
    System.out.println(numBreaks);
    pageBreaks = new int[numBreaks];
    for (int b = 0; b < numBreaks; b++) {
        pageBreaks[b] = b * linesPerPage;
    }
}

if (pageIndex > pageBreaks.length) {
    return NO_SUCH_PAGE;
}

Graphics2D g2D = (Graphics2D) graphics;
if (pageIndex == 0) {
    g2D.translate( (int) pageFormat.getImageableX() + (int) pageFormat.getImageableWidth() / 2 ,
        (int) pageFormat.getImageableY() + 60 + (int) (pageFormat.getImageableHeight() * 0.25f));
}
else {
    g2D.translate(pageFormat.getImageableX(), pageFormat.getImageableY());
}
//int start = (pageIndex == 1) ? 0 : pageBreaks[pageIndex - 1];
int start = pageBreaks[pageIndex - 1];
int end = (pageIndex == pageBreaks.length) ? textLines.length : pageBreaks[pageIndex];
for (int line = start; line < end; line++) {
    y += lineHeight;
    System.out.println(y);
    graphics.drawString(textLines[line], (int) 0, y);
}
```

Демонстрация работы программы



```
package com.company;  
  
import java.awt.*;  
import java.awt.geom.AffineTransform;  
import java.awt.geom.PathIterator;  
import java.awt.geom.Point2D;  
import java.awt.geom.Rectangle2D;  
import java.util.ArrayList;  
import java.util.Comparator;  
  
public class AstroidShape implements Shape {
```

```

point2DS.clear();
for (float t=Tmin;t<Tmax;t+=jump){
    int x = GetX(t);
    int y = GetY(t);
    g.fillRect(x,y,2,2);
    point2DS.add(new Point(x,y));
}

point2DS.sort(comparator);
int yMin = point2DS.get(0).y;
result=new ArrayList<Point>();

for(int i=0;i<point2DS.size();i++){
    if(point2DS.get(i).y<=yMin){
        result.add(point2DS.get(i));
    }
}
point2DS.clear();
int prevSize =result.size();
for(int i =prevSize-1;i>=0;i--){
    result.add(new Point(result.get(i).x,2*yMin- result.get(i).y));
}
}

public void UpdateList(){
    for (float t=Tmin;t<Tmax;t+=jump){

```

```

        int x = GetX(t);
        int y = GetY(t);
        point2DS.add(new Point(x,y));
    }

    point2DS.sort(comparator);
    int yMin = point2DS.get(0).y;
    result=new ArrayList<Point>();

    for(int i=0;i<point2DS.size();i++){
        if(point2DS.get(i).y<=yMin){
            result.add(point2DS.get(i));
        }
    }
    point2DS.clear();
    int prevSize =result.size();
    for(int i =prevSize-1;i>=0;i--){
        result.add(new Point(result.get(i).x,2*yMin- result.get(i).y));
    }
}

```


Задание №5

Условие задачи

Разработайте систему классов/интерфейсов для предметной области Вашего варианта задания. Данные необходимо упорядочить по атрибутам/свойствам товаров, предметов и т.п. в виде дерева.

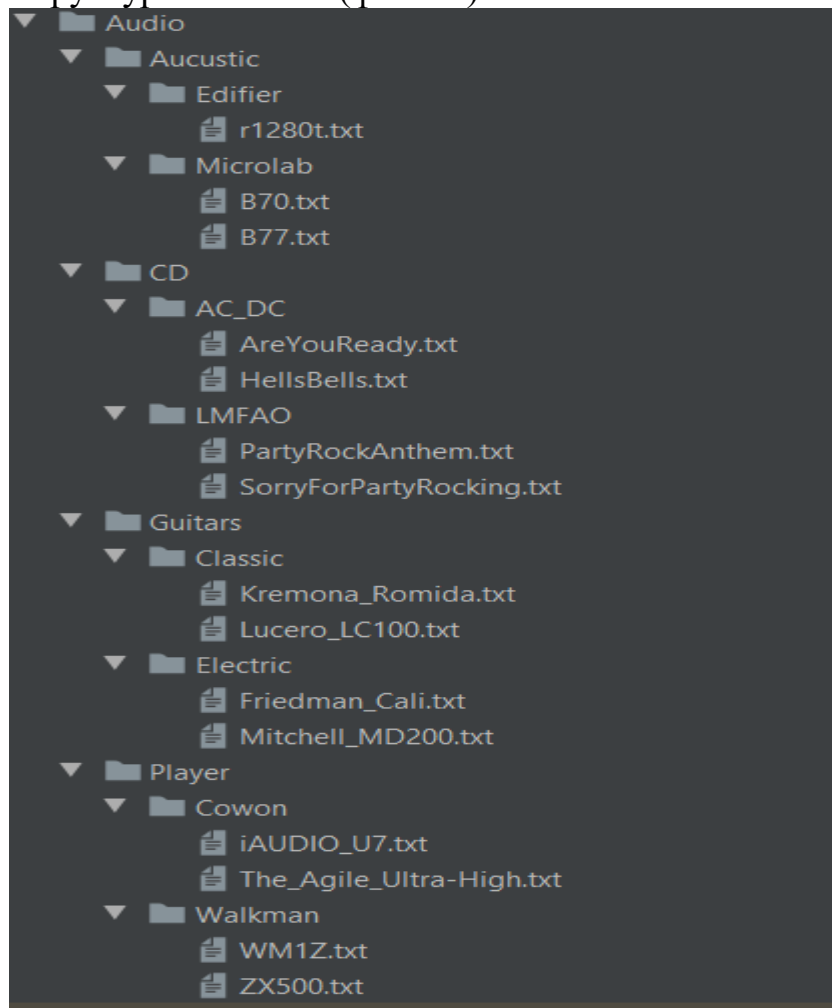
- Разработайте графическое приложение для ввода/отображения данных Вашего варианта задания. При отображении структуры данных в виде дерева реализуйте интерфейс *javax.swing.Tree.TreeModel*. Листья дерева отображайте в виде таблицы, для этого реализуйте интерфейс *javax.swing.table.TableModel*. (пример похожего приложения – Проводник Windows)

- При реализации интерфейса следуйте рекомендациям стандарта CUI (Common User Interface).

Аудио – магазин.

Особенности реализации

Структура магазина (файлов)



Для отображения файлов я сделал класс наследуемый от *AbstractTableModel*

Так же был создан вспомогательный класс DynamicTree для работы с данными. Его настройка происходила следующим образом:

```
String p1Name = "Aucustic";
String p2Name = "CD";
String p3Name = "Guitars";
String p4Name = "Player";

String c1_1Name = "Edifier";
String c1_2Name = "Microlab";

String c2_1Name = "AC_DC";
String c2_2Name = "LMFAO";

String c3_1Name = "Classic";
String c3_2Name = "Electric";

String c4_1Name = "Cowon";
String c4_2Name = "Walkman";

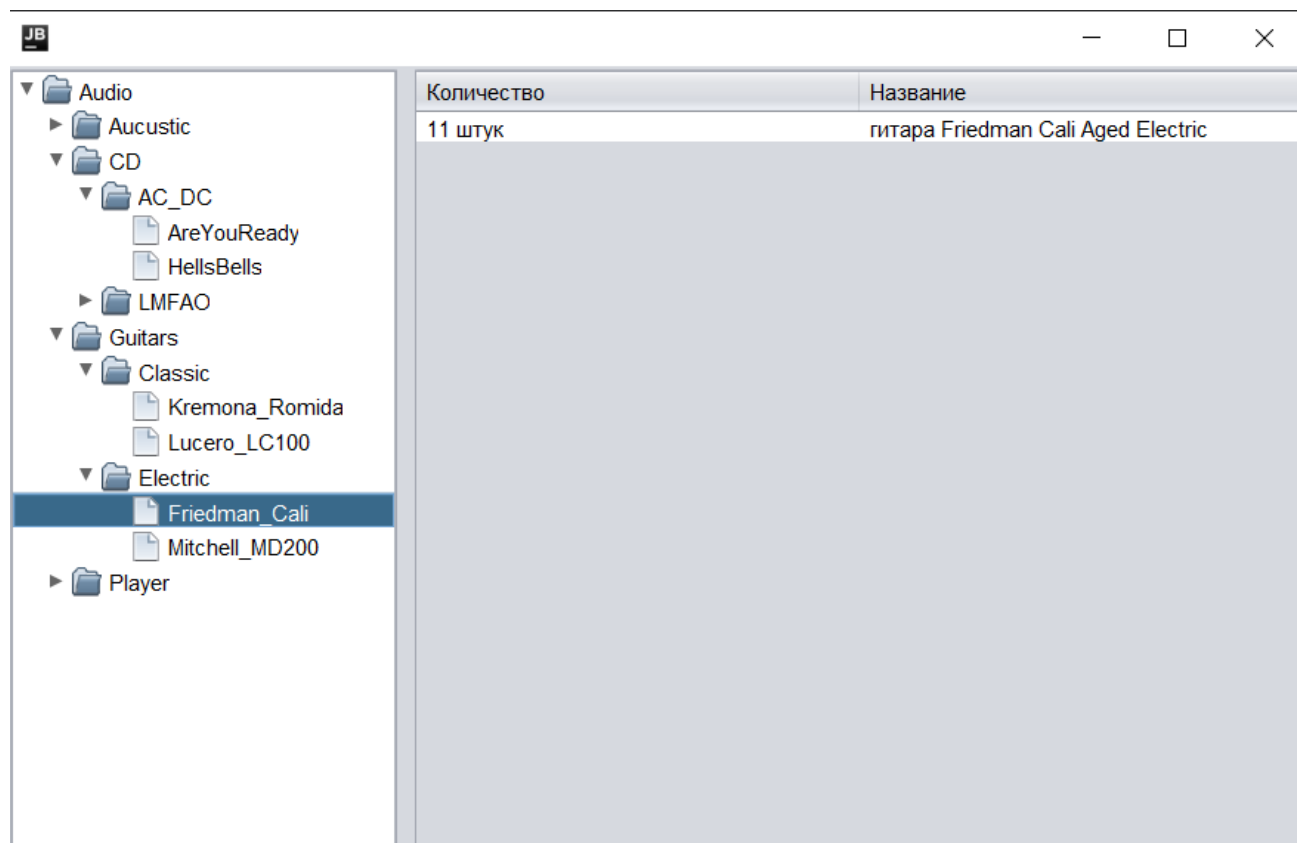
p1 = this.addObject( parent: null, p1Name);
p2 = this.addObject( parent: null, p2Name);
p3 = this.addObject( parent: null, p3Name);
p4 = this.addObject( parent: null, p4Name);

c1_1 = this.addObject(p1, c1_1Name);
c1_2 = this.addObject(p1, c1_2Name);
processFilesFromFolder( path: "Audio\\Aucustic\\Edifier\\", c1_1);
processFilesFromFolder( path: "Audio\\Aucustic\\Microlab\\", c1_2);

c2_1 = this.addObject(p2, c2_1Name);
c2_2 = this.addObject(p2, c2_2Name);
processFilesFromFolder( path: "Audio\\CD\\AC_DC\\", c2_1);
processFilesFromFolder( path: "Audio\\CD\\LMFAO\\", c2_2);

c3_1 = this.addObject(p3, c3_1Name);
```

Демонстрация работы программы



Задание №6

Условие задачи

Для выполнения задания используется ваш вариант решения задания №3. Модифицируйте вашу программу следующим образом.

Создайте тестовое приложение, добавьте в ваш класс рисования алгебраической линии возможность «перетаскивание» (drag-and-drop). Реализуйте необходимые интерфейсы в классе и в приложении для демонстрации «перетаскивания» алгебраической линии между несколькими копиями тестового приложения.

Особенности реализации

Под основу была взята работа из задания №3, реализованы интерфейсы:

```
DragGestureListener, // For recognizing the start of drags
DragSourceListener,  // For processing drag source events
DropTargetListener,  // For processing drop target events
MouseListener,        // For processing mouse clicks
MouseMotionListener  // For processing mouse drags
```

Так же для клонирования рисунка был реализован интерфейс clonable

```

@Override
public Object clone() {
    try {
        return super.clone();
    } catch (CloneNotSupportedException e) {
        System.out.println("Couldnt clone");// This should never happen
        return this;
    }
}

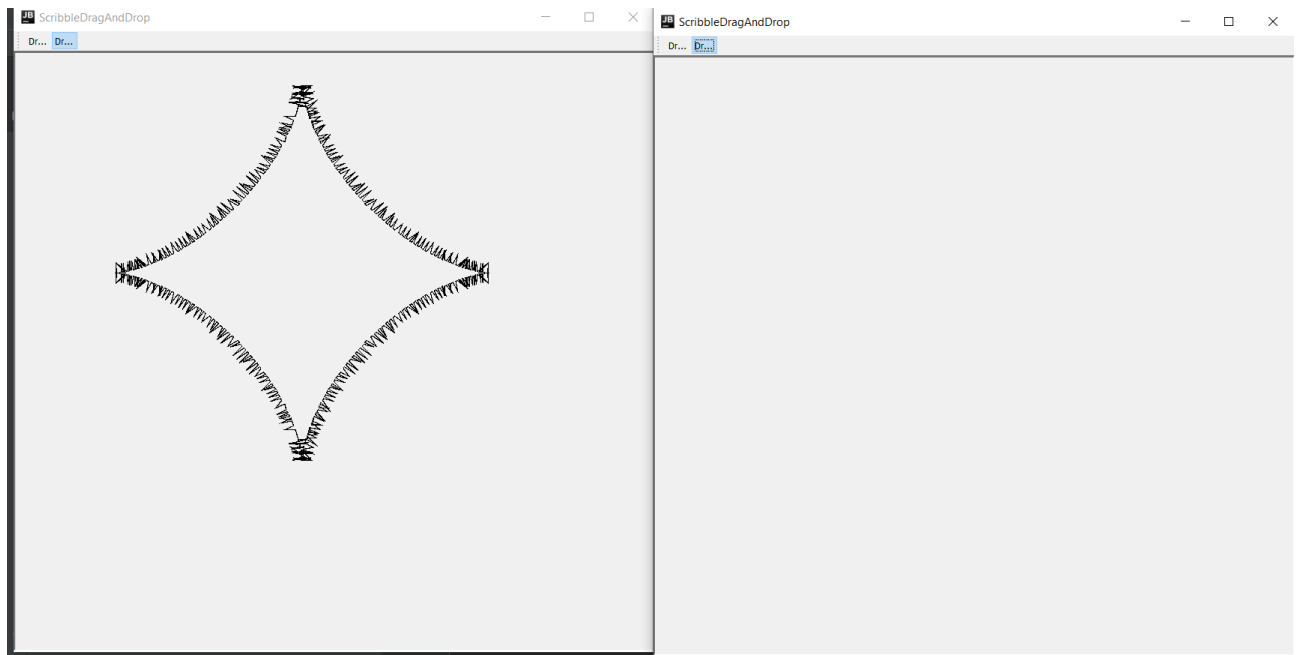
void translate(int x, int y) {
    centerX += x;
    centerY += y;
}

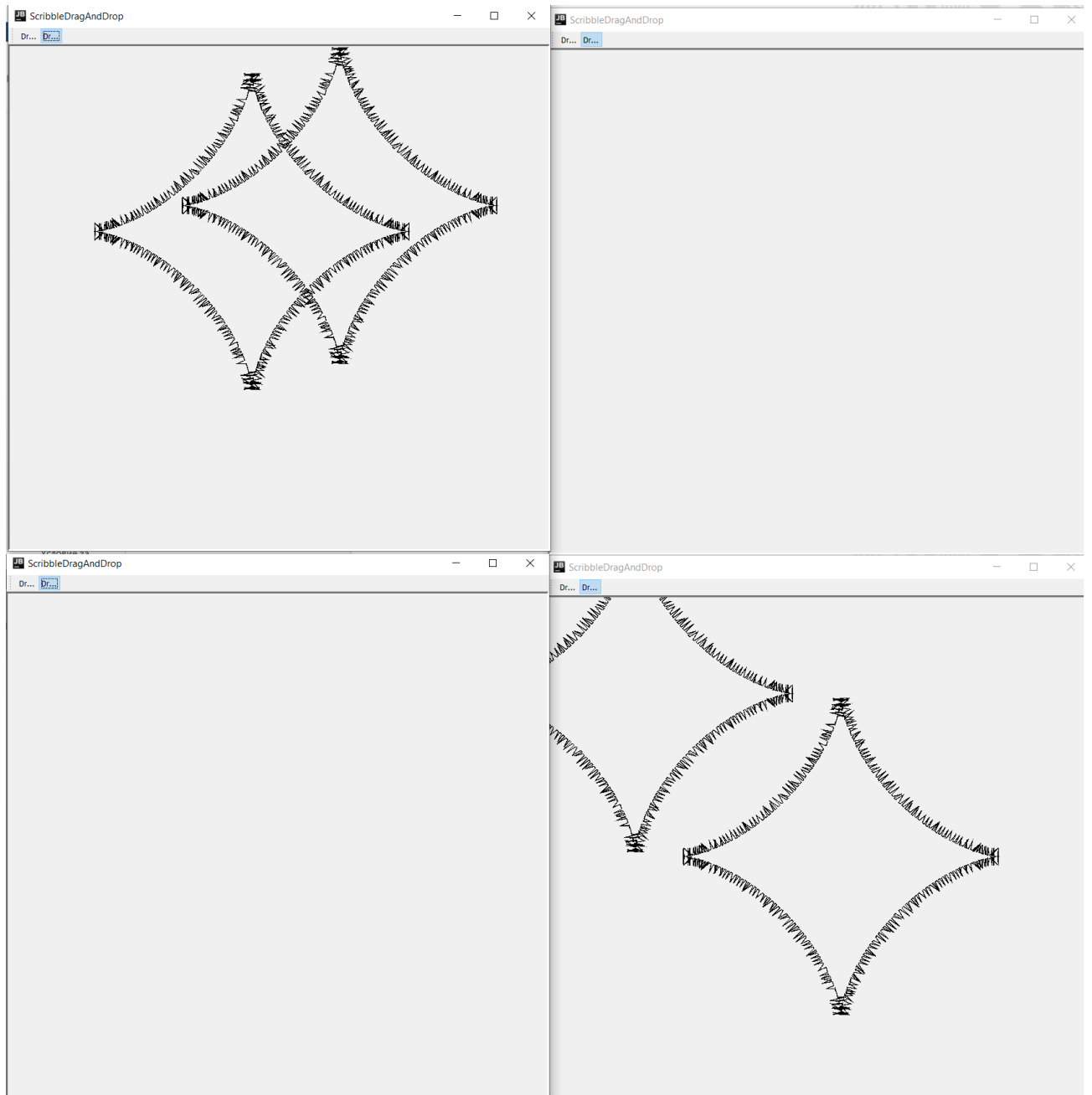
@Override
public String toString() { return this.centerX + " " + this.centerY ; }

static AstroidShape getFromString(String line) {
    String[] arr = line.split( regex: " ");
    return new AstroidShape(Integer.parseInt(arr[0]), Integer.parseInt(arr[1]));
}

```

Демонстрация работы программы





Задание №7

Условие задачи

- Исследовать предложенную предметную область, спроектировать структуру базы данных объектов выбранной предметной области (из не менее чем 2-х таблиц объектов).

- Разработайте графическое приложение для создания/ввода/отображения БД Вашего варианта задания. Содержимое БД отображайте в виде таблиц. - При реализации интерфейса следуйте рекомендациям стандарта CUI (Common User Interface).

АудиоМагазин

Особенности реализации

Работа идет с SQLite. Создали две таблицы для описания типа, и самих товаров. Работаю с БД при помощи следующих классов:

```
private static Connection connection;  
private static Statement statement;  
private static ResultSet resultSet;
```

Фрагмент кода: подключение к БД

```
private static void connectionDB() {  
    try {  
        connection = null;  
        Class.forName("org.sqlite.JDBC");  
        connection = DriverManager.getConnection( url: "jdbc:sqlite:Audioshop.db");  
  
        statement = connection.createStatement();  
        String selectSQL = "SELECT * FROM Audios";  
        resultSet = statement.executeQuery(selectSQL);  
  
        while (resultSet.next()) {  
            Audio Audio = new Audio(resultSet.getString( s: "name"),  
                                    resultSet.getString( s: "Age"),  
                                    resultSet.getString( s: "type"),  
                                    resultSet.getString( s: "ingTime"),  
                                    resultSet.getString( s: "amount"));  
            AudiosData.add(Audio);  
        }  
  
        selectSQL = "SELECT * FROM audiotypes";  
        resultSet = statement.executeQuery(selectSQL);  
  
        while (resultSet.next()) {  
            audiotype audioamount = new audiotype(  
                resultSet.getString( s: "type"),  
                resultSet.getString( s: "id"),  
                resultSet.getString( s: "description")  
            );  
        }  
    }  
}
```


Демонстрация работы программы

Audioshop table

Name	Age	type	amount
ACDC	23	CD	233
JackS	2	CD	13
23	12	CD	3
Rock2V0	12	Guitar	45

<input type="text" value="Name"/>	<input type="text" value="Age"/>	<input type="text" value="Guitar"/>	<input type="text" value="Age"/>	<input type="text" value="amount"/>	<input type="button" value="Add audio"/>
-----------------------------------	----------------------------------	-------------------------------------	----------------------------------	-------------------------------------	--

type	type description
CD	SuperMusic
Guitar	GuitarStuff

Задание №8

Условие задачи

1) Изучите материал примера по быстрому введению в среду разработки NetBeans и компоненты JavaBeans по адресу: <http://docs.oracle.com/javase/tutorial/javabeans/quick/index.html>

2) Разработайте простой компонент вашего варианта задания на базе класса Canvas. Создайте файл манифеста и упакуйте компонент вместе с исходным кодом разработанных классов. При разработке поместите все ваши классы в пакет: bsu.fpmi.educational_practice

3) Создайте тестовое приложение в NetBeans с использованием вашего компонента.

Символ «вопрос». Свойства: диаметр, цвет фона, цвет символа

Особенности реализации

За рисование отвечает класс Info, который унаследован от Canvas.

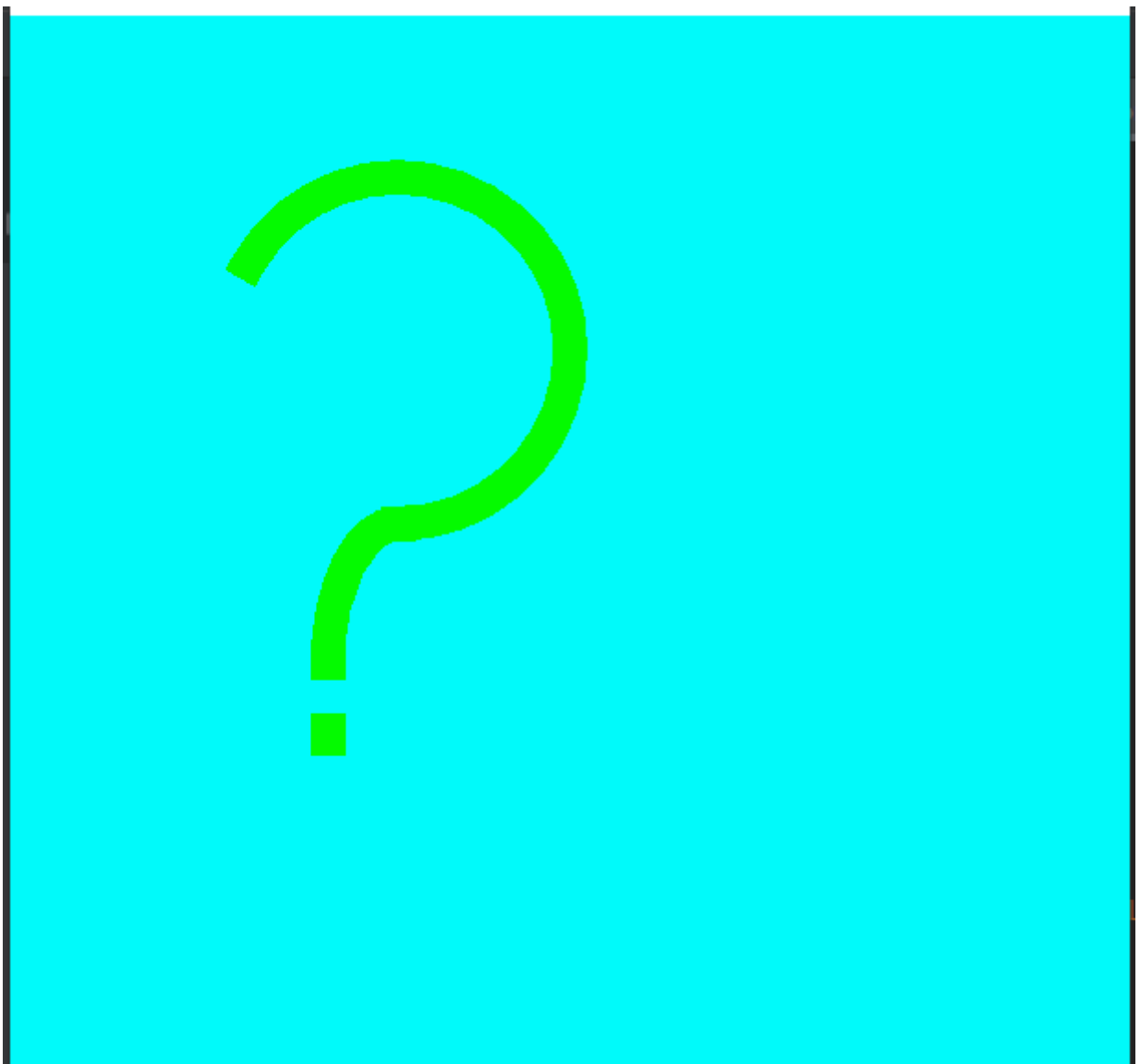
Данный класс предоставляет следующий интерфейс для редактирования своих свойств:

```
public Color getIColor() {return this.iColor;}
public Color getBColor() {return this.bColor;}
public float getDiam() {return this.diam;}

public void setIColor(Color tmp) { this.iColor = tmp; }
public void setBColor(Color tmp) { this.bColor = tmp; }
public void setDiam(float tmp) { this.diam = tmp; }

public void setX(int x){this.x = x;}
public void setY(int y){this.y = y;}
```

Демонстрация работы программы



Задание №9

Условие задачи

1) Разработайте компонент вашего варианта задания. Создайте файл манифеста и упакуйте компонент вместе с исходным кодом разработанных классов. При разработке поместите все ваши классы в пакет: `bsu.fpmi.educational_practice`

2) Компонент должен реализовать класс `BeanInfo` с информацией о компоненте.

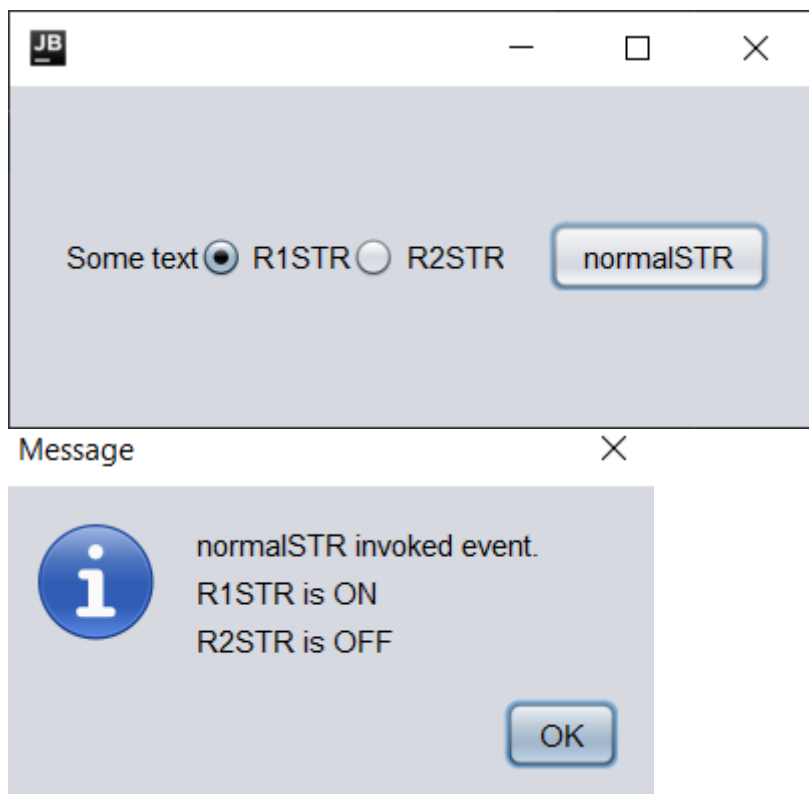
3) Создайте тестовое приложение в NetBeans с использованием вашего компонента.

Однострочный статический текст, две строки ввода и кнопка-флаг. Свойства: текст, текст флага, символ подтверждения. Событие генерируется при вводе символа подтверждения. Событие передаёт ещё состояние флага и содержимое строк ввода.

Особенности реализации

За работу с UI отвечает класс `NewJPanel`, который содержит все необходимые методы для рисования рабочего окна. Для обработки событий были создан класс `АсептEvent` и интерфейс для подписчиков - `АсептListener`. Для хранения информации об `JavaBean` был создан класс `BeanInfo`

Демонстрация работы программы



Задание №10

Условие задачи

Для решения задания №10 используем решённый вариант задания №9. Номера заданий сохраняются. Модифицируем тип свойства компонента так, как указано ниже: Основная задача: Создаём собственный редактор для каждого свойства компонента. Каждый редактор ограничивает возможные значения свойства, предоставляя выбор из списка трёх – пяти допустимых значений (т. е. определяем методы `getTags()`). Регистрируем редакторы в классе `BeanInfo` компонента.

Однострочный статический текст, две строки ввода и кнопка-флаг. Свойства: текст, текст флага, символ подтверждения. Событие генерируется при вводе символа подтверждения. Событие передаёт ещё состояние флага и содержимое строк ввода

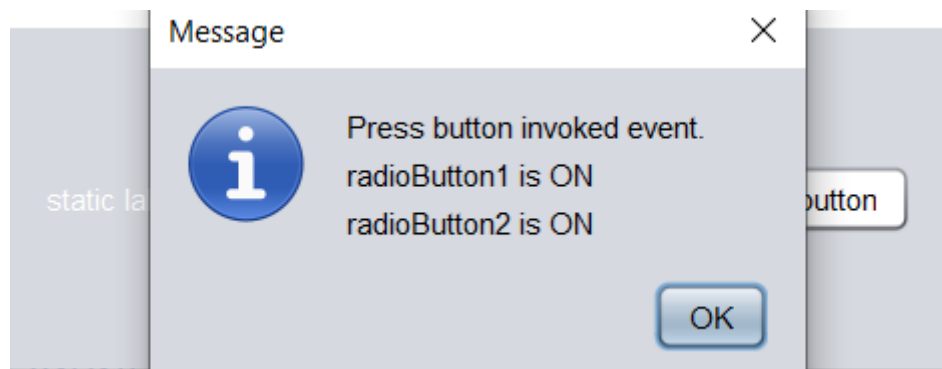
Особенности реализации

Для редактирования свойств кнопок и текста были созданы классы

`ButtonPropertyEditor`, `LabelPropertyEditor`, `RadioPropertyEditor`. Все они наследуются от класса `PropertyEditorSupport` и переопределяют его метод `GetTags()` для получения информации о возможных конфигурациях соответствующих компонентов.

Демонстрация работы программы





Задание №11

Условие задачи

Создать сервлет и взаимодействующие с ним пакеты Java-классов и HTML документов, выполняющие действия для решения вашего варианта задания. Представить решение в виде web-приложения (как в примере).

Вариант 6: Вывод фрагментов текстов шрифтами различного размера. Размер шрифта и количество строк задается на стороне клиента.

Особенности реализации

Основной функция для связи между сервером и нашим сервлетом является функция doGet(). Данный метод вызывается всегда пока открыто окно в браузере и принимает информацию о странице с сервера.

Был создан web.xml файл, для mapping между index.html и классами java.

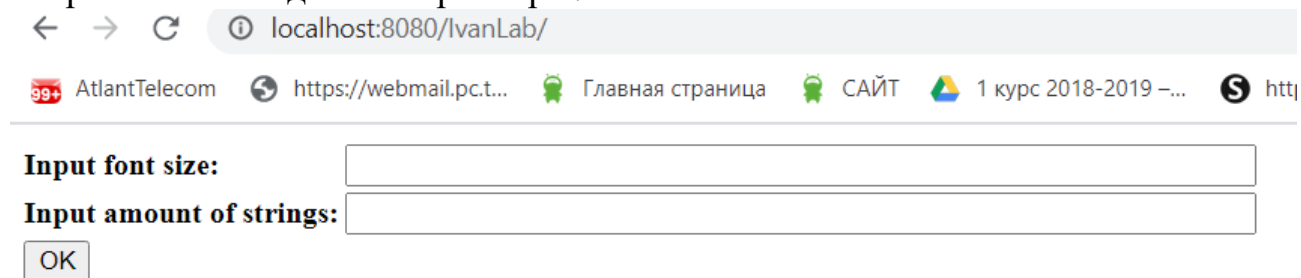
Текст для предоставления пользователю выбирался из заранее подготовленного текста. Использовался сервер tomcat 8.5.

Реализация предоставления текста:

```
List<String> list_ = new ArrayList<>();
InputStream input = getServletContext().getResourceAsStream(s: "file.txt");
BufferedReader buf = new BufferedReader(new InputStreamReader(input));
String strLine;
StringBuilder stringBuilder = new StringBuilder();
int i = 0;
while ((strLine = buf.readLine())!=null) {
    list_.add(strLine);
    i++;
}
if (amount>i)
    req.setAttribute(s: "error", o: "Not enough strings in file");
else{
    req.setAttribute(s: "text", list_);
    req.setAttribute(s: "size", font_size);
    req.setAttribute(s: "amount", amount);
}
req.getRequestDispatcher(s: "Output.jsp").forward(req, resp);
}
```

Демонстрация работы программы

Запрос на необходимые параметры:



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/IvanLab/'. Below the address bar, there are several tabs: 'AtlantTelecom', 'https://webmail.pc.t...', 'Главная страница', 'САЙТ', and '1 курс 2018-2019 -...'. The main content area contains a form with two input fields. The first field is labeled 'Input font size:' and the second field is labeled 'Input amount of strings:'. Below these fields is an 'OK' button.

← → ↻ ⓘ localhost:8080/IvanLab/

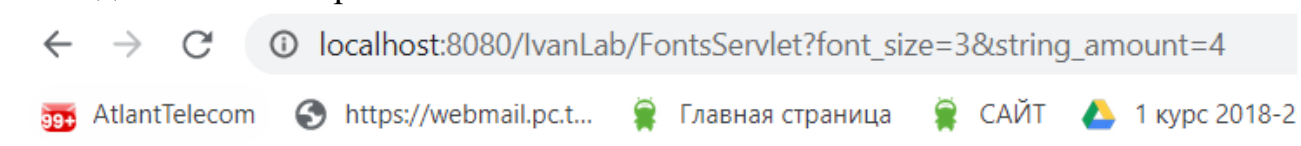
AtlantTelecom https://webmail.pc.t... Главная страница САЙТ 1 курс 2018-2019 -... http

Input font size:

Input amount of strings:

OK

Вывод ответа на запрос



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/IvanLab/FontsServlet?font_size=3&string_amount=4'. Below the address bar, there are several tabs: 'AtlantTelecom', 'https://webmail.pc.t...', 'Главная страница', 'САЙТ', and '1 курс 2018-2019 -...'. The main content area is empty.

← → ↻ ⓘ localhost:8080/IvanLab/FontsServlet?font_size=3&string_amount=4

AtlantTelecom https://webmail.pc.t... Главная страница САЙТ 1 курс 2018-2019 -...

I

am

student

of

Задание №12

Условие задачи

- 1) Изучите пример 2
- 2) Проанализируйте ваш вариант задания. Можно ли его реализовать как часть MUD системы (например, в одной из комнат MudPlace), требуется ли для этого внести изменения в парадигму MUD? Какие изменения потребует реализация клиента MUD, другие классы примера? Оформите эти размышления в вашем отчёте в качестве анализа предметной области. При реализации, по возможности, используйте парадигму MUD и классы примера 2 при реализации вашего варианта задания.
- 3) Создайте на основе технологии RMI клиент/серверное приложение:
Игра в крестики-нолики по сети. Игра между двумя клиентами через сервер. Клиент предлагает поиграть и ждёт, пока другой клиент согласится на игру. Сервер организует связь между играющими клиентами

Особенности реализации

Основой в работе RMI является интерфейс, который наследуется от интерфейса Remote и необходим для определения функций которыми может пользоваться клиент на своей стороне.

В моем случае для игры я ограничился следующим интерфейсом

```
public interface RemoteGameServer extends Remote {  
  
    boolean GameFound(String forUser) throws RemoteException;  
  
    boolean myGameFinished(String forUser) throws RemoteException;  
  
    String getCurrentStepData(String playerName) throws RemoteException;  
  
    boolean isNowCurrentStepOf(String playerName) throws RemoteException;  
  
    void placeSymbol(Pair<Integer, Integer> point, String senderName) throws RemoteException;  
  
    boolean winner(String userName) throws RemoteException;  
  
    boolean isDraw(String playerName) throws RemoteException;  
  
}
```

Демонстрация работы программы

Подключение клиента:

```
Logging....  
Enter your new Game name: |
```

Поиск игры

```
Logging....  
Enter your new Game name: magtar  
Logged OK!  
Would you like to find a Game? y/n: y  
Searching a Game for you. Please, wait...|
```

Начало игры

```
Number of steps:    0  
MAP:  
  
    1  2  3  
1   .  .  .  
2   .  .  .  
3   .  .  .  
  
Your symbol:  
X
```

Конец игры

Number of steps: 5

MAP:

	1	2	3
1	X	X	X
2	.	0	.
3	0	.	.

Your symbol:

X

Congratulations: you won!

