**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](#_Toc41327236)

[Условие задачи 4](#_Toc41327237)

[Особенности реализации 4](#_Toc41327238)

[Демонстрация работы программы 6](#_Toc41327239)

[Задание №2 7](#_Toc41327240)

[Условие задачи 7](#_Toc41327241)

[Особенности реализации 7](#_Toc41327242)

[Демонстрация работы программы 8](#_Toc41327243)

[Задание №3 9](#_Toc41327244)

[Условие задачи 9](#_Toc41327245)

[Особенности реализации 9](#_Toc41327246)

[Демонстрация работы программы 12](#_Toc41327247)

[Задание №4 13](#_Toc41327248)

[Условие задачи 13](#_Toc41327249)

[Особенности реализации 13](#_Toc41327250)

[Демонстрация работы программы 15](#_Toc41327251)

[Задание №5 17](#_Toc41327252)

[Условие задачи 17](#_Toc41327253)

[Особенности реализации 17](#_Toc41327254)

[Демонстрация работы программы 19](#_Toc41327255)

[Задание №6 20](#_Toc41327256)

[Условие задачи 20](#_Toc41327257)

[Особенности реализации 20](#_Toc41327258)

[Демонстрация работы программы 22](#_Toc41327259)

[Задание №7 24](#_Toc41327260)

[Условие задачи 24](#_Toc41327261)

[Особенности реализации 24](#_Toc41327262)

[Демонстрация работы программы 25](#_Toc41327263)

[Задание №8 25](#_Toc41327264)

[Условие задачи 25](#_Toc41327265)

[Особенности реализации 25](#_Toc41327266)

[Демонстрация работы программы 26](#_Toc41327267)

[Задание №9 27](#_Toc41327268)

[Условие задачи 27](#_Toc41327269)

[Особенности реализации 27](#_Toc41327270)

[Демонстрация работы программы 28](#_Toc41327271)

[Задание №10 29](#_Toc41327272)

[Условие задачи 29](#_Toc41327273)

[Особенности реализации 29](#_Toc41327274)

[Демонстрация работы программы 29](#_Toc41327275)

[Задание №11 31](#_Toc41327276)

[Условие задачи 31](#_Toc41327277)

[Особенности реализации 31](#_Toc41327278)

[Демонстрация работы программы 31](#_Toc41327279)

[Задание №12 34](#_Toc41327280)

[Условие задачи 34](#_Toc41327281)

[Особенности реализации 34](#_Toc41327282)

[Демонстрация работы программы 35](#_Toc41327283)

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

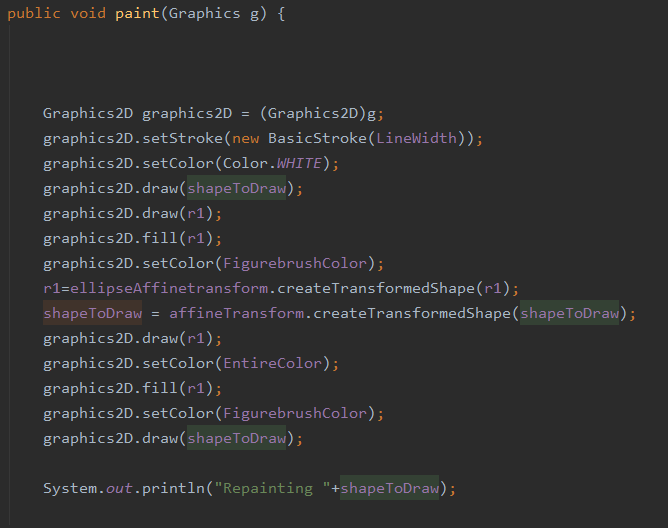
- выполнить рисунок в окне фрейма с выбранной толщиной границы фигуры, цветом

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

ваших программ).

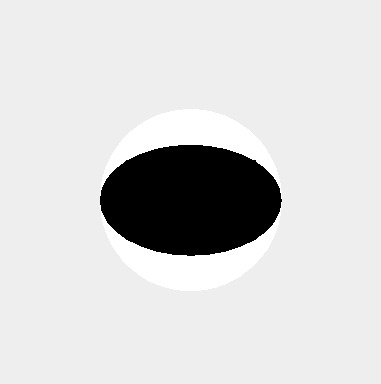
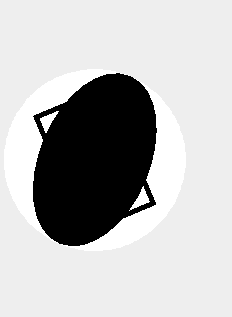
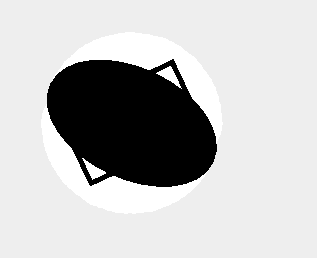
## Особенности реализации

Обновление афинных преобразований рекурентно исходя из предыдущих при каждой перерисовке:



Перерисовка происходит каждые 120 ms в отдельном потоке.

## Демонстрация работы программы



# Задание №2

## Условие задачи

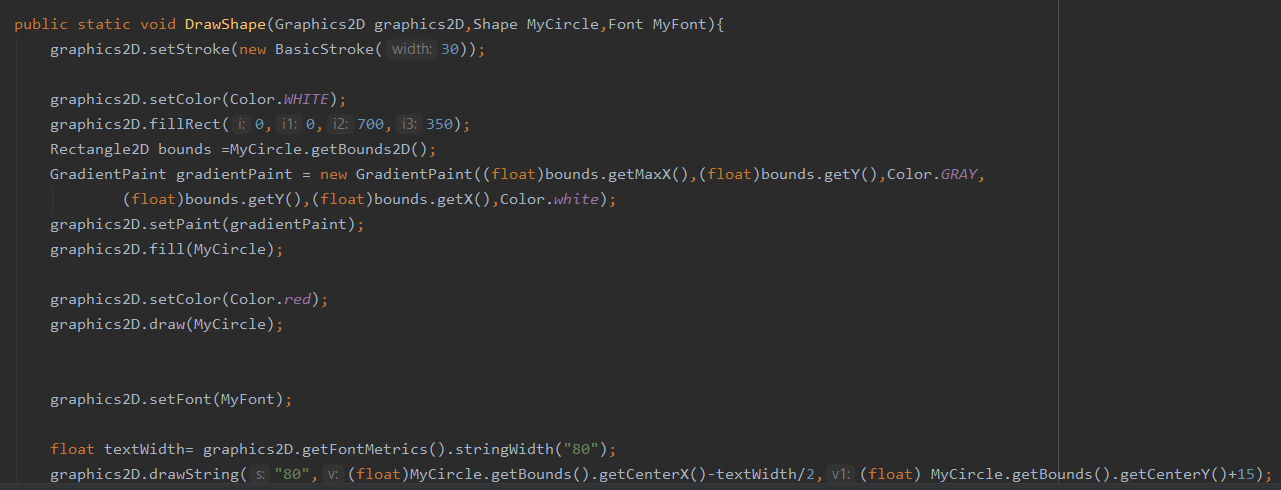
Создайте тестовое приложение (Frame/JFrame) для демонстрации вашего решения, при этом: - для изображения указанной в задании фигуры создать класс, реализующий интерфейс Shape; - создайте указанный фильтр изображения; при тестировании выведите фигуру без фильтра и с фильтром (аналогично фильтрам из примеров); - 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

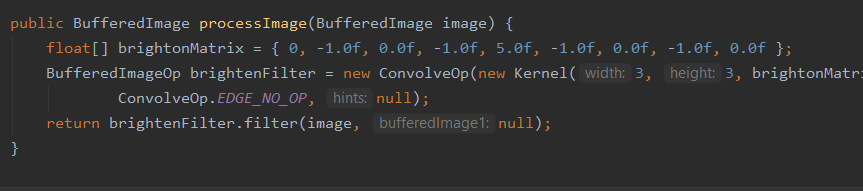
Фильтр: Brighten

## Особенности реализации

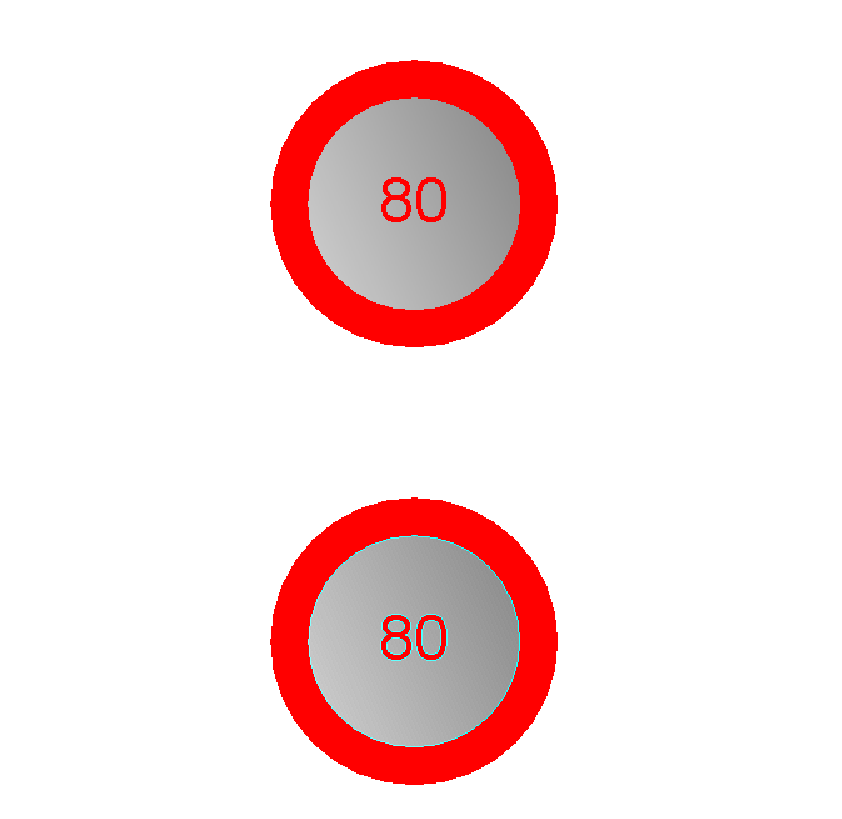
Рисование изображения



Накладывание фильтра на изображение.



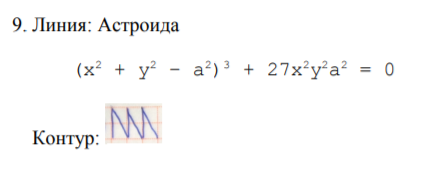
## Демонстрация работы программы



# Задание №3

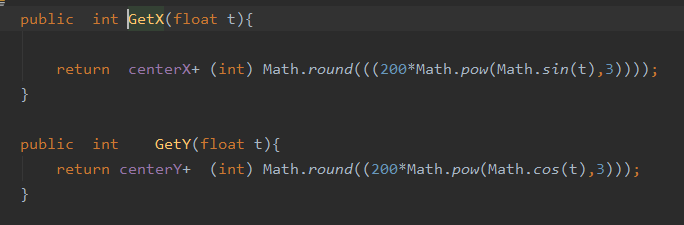
## Условие задачи

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.

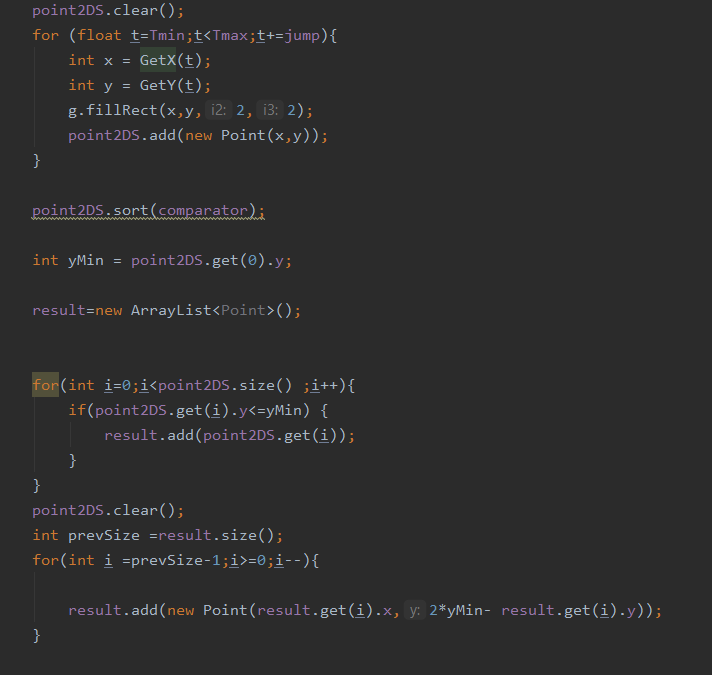


## Особенности реализации

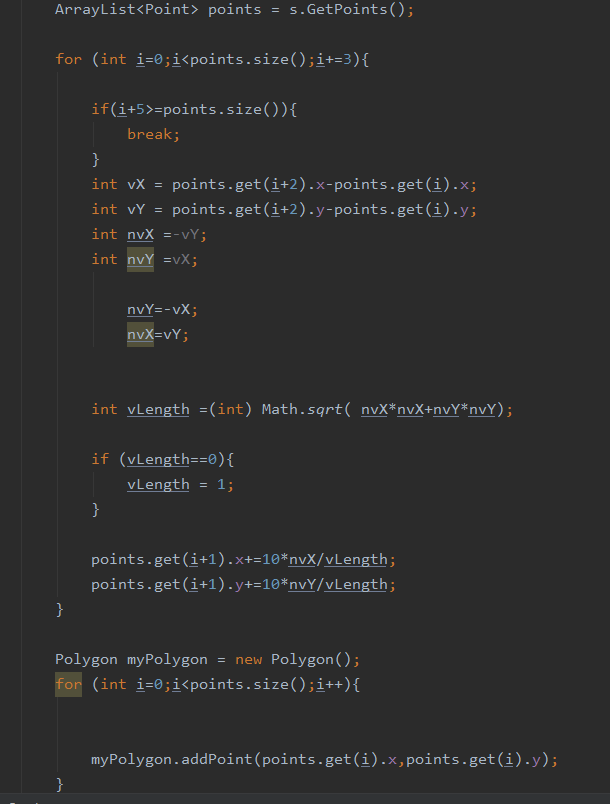
Чтобы удобно рисовать фигуру будем работать с параметрическим видом данной функции.



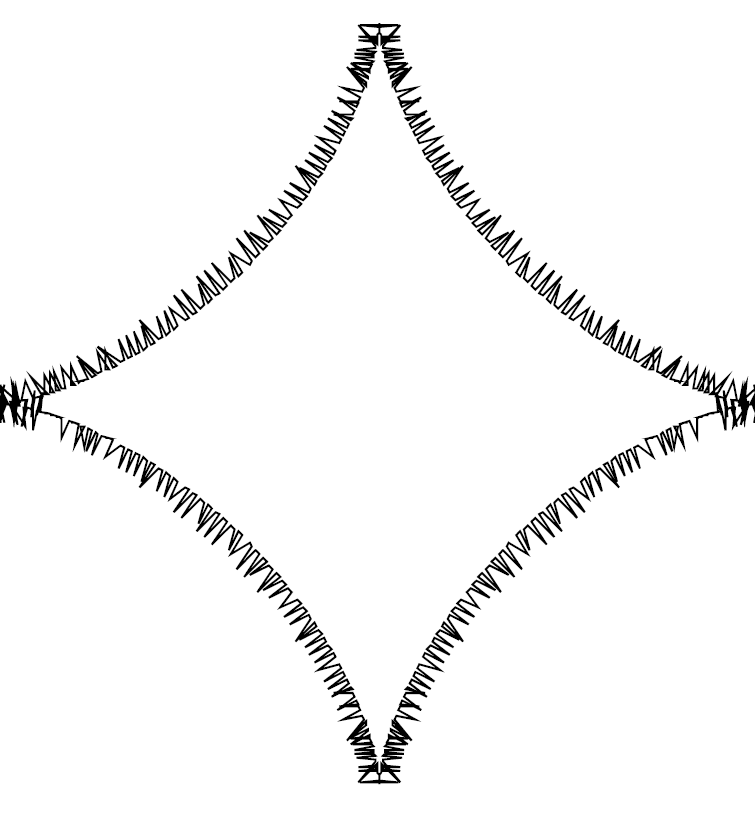
Рисование каждой точки будем делать с определенным шагом.



Отрисовываем контур в своем классе MyStroke



## 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.

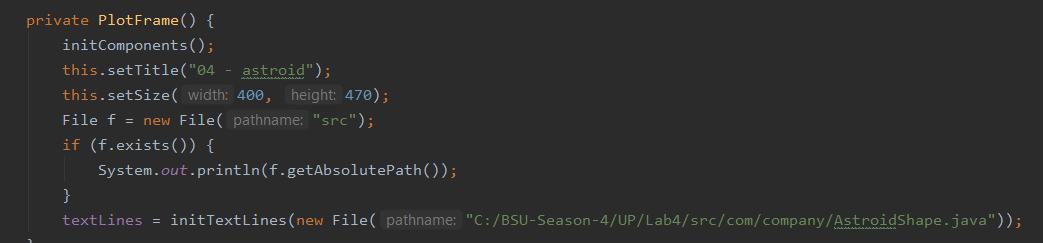
Рисунок должен занимать не более половины страницы, при печати выровнять его по горизонтали.

## Особенности реализации

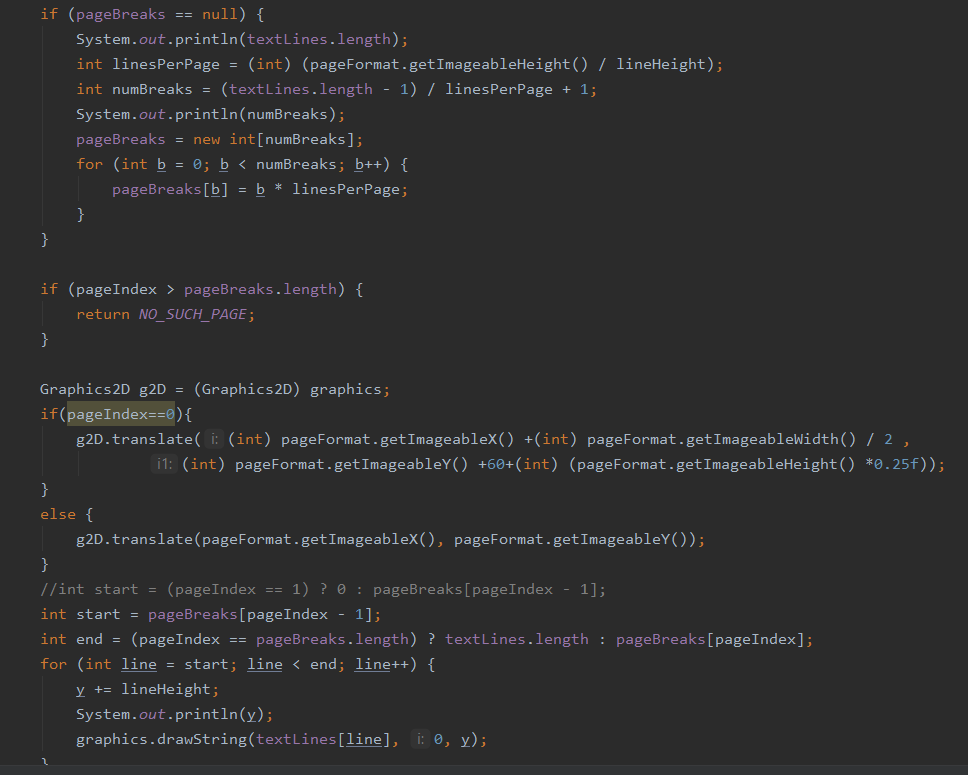
Для печати был создан класс PlotFrame, который реализует интерфейс

Printable

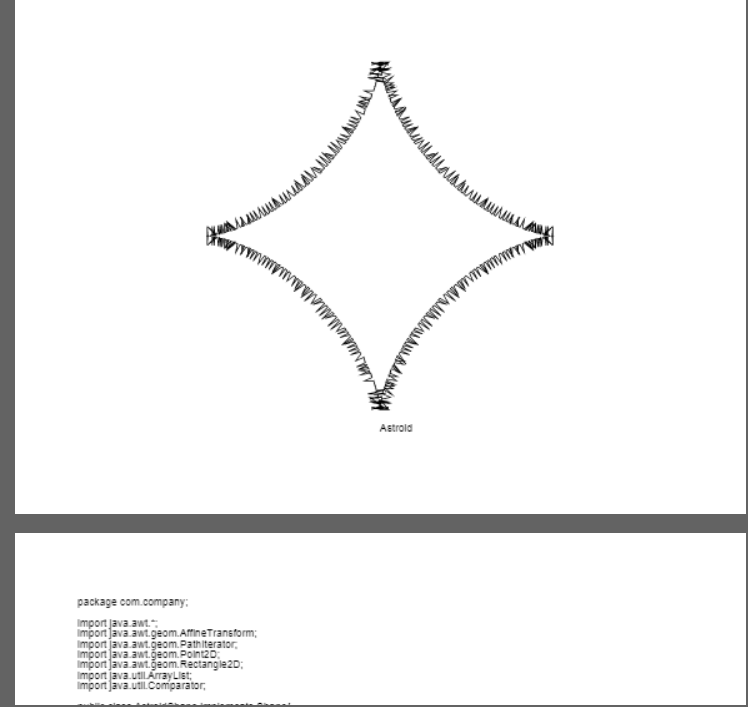
Получаем текст кода:

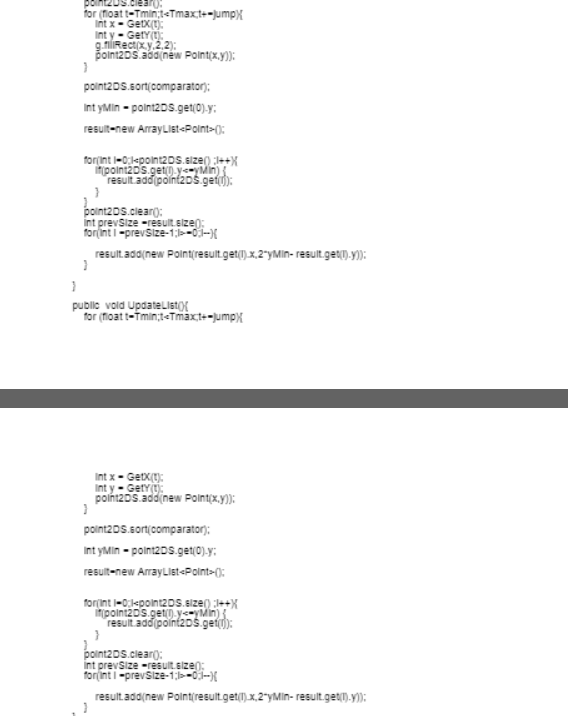


Рассчитываем разметку страницу по количеству строк и их размера.



## Демонстрация работы программы





# 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.

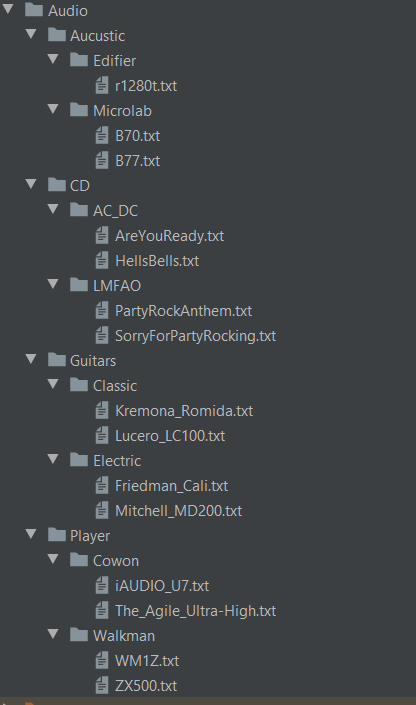
- Разработайте графическое приложение для ввода/отображения данных Вашего варианта задания. При отображении структуры данных в виде дерева реализуйте интерфейс *javax.swing.Tree.TreeModel*. Листья дерева отображайте в виде таблицы, для этого реализуйте интерфейс *javax.swing.table.TableModel.* (пример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

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

Аудио – магазин.

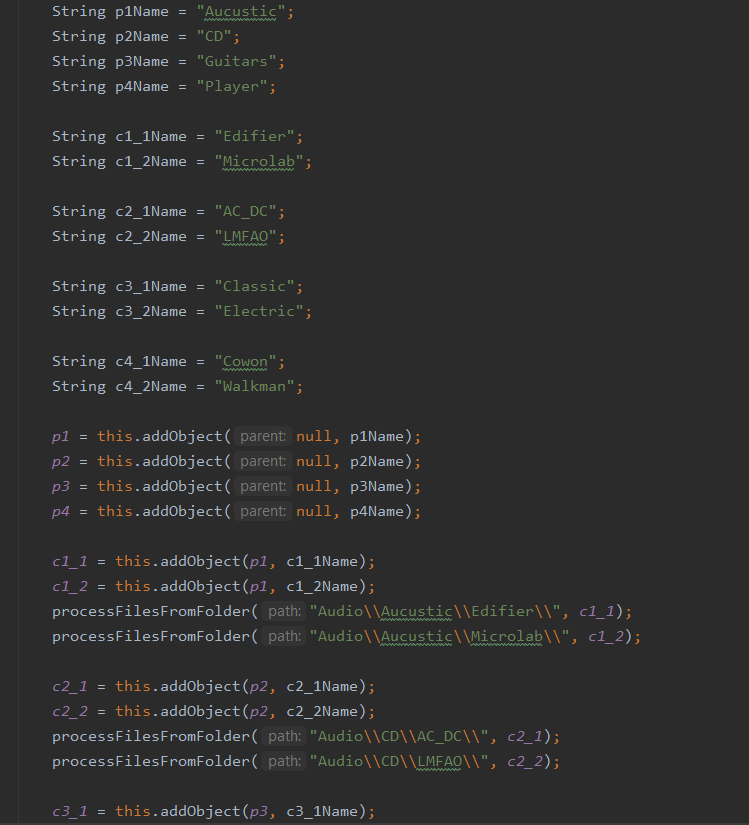
## Особенности реализации

Структура магазина (файлов)

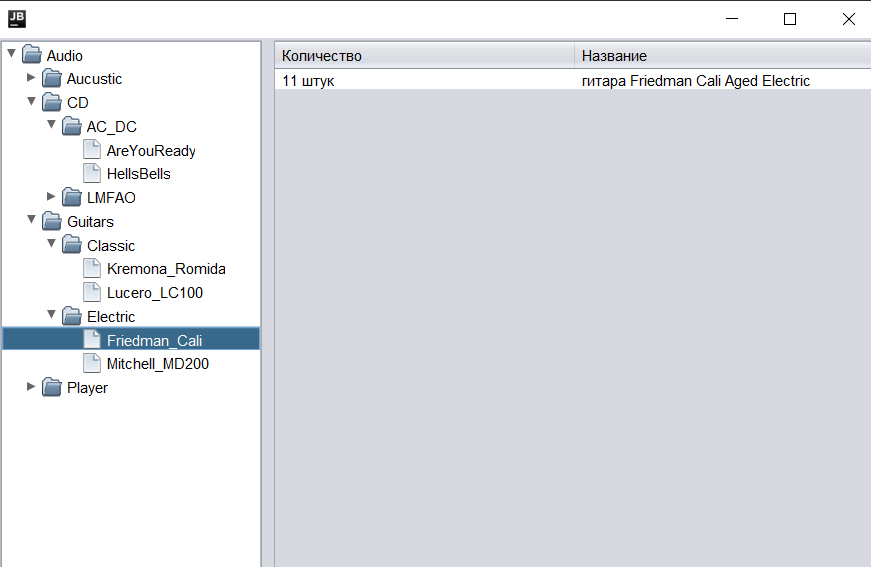


Для отображения файлов я сделал класс наследуемый от AbstractTableModel

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



## Демонстрация работы программы



# Задание №6

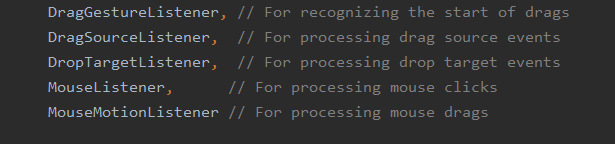
## Условие задачи

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.

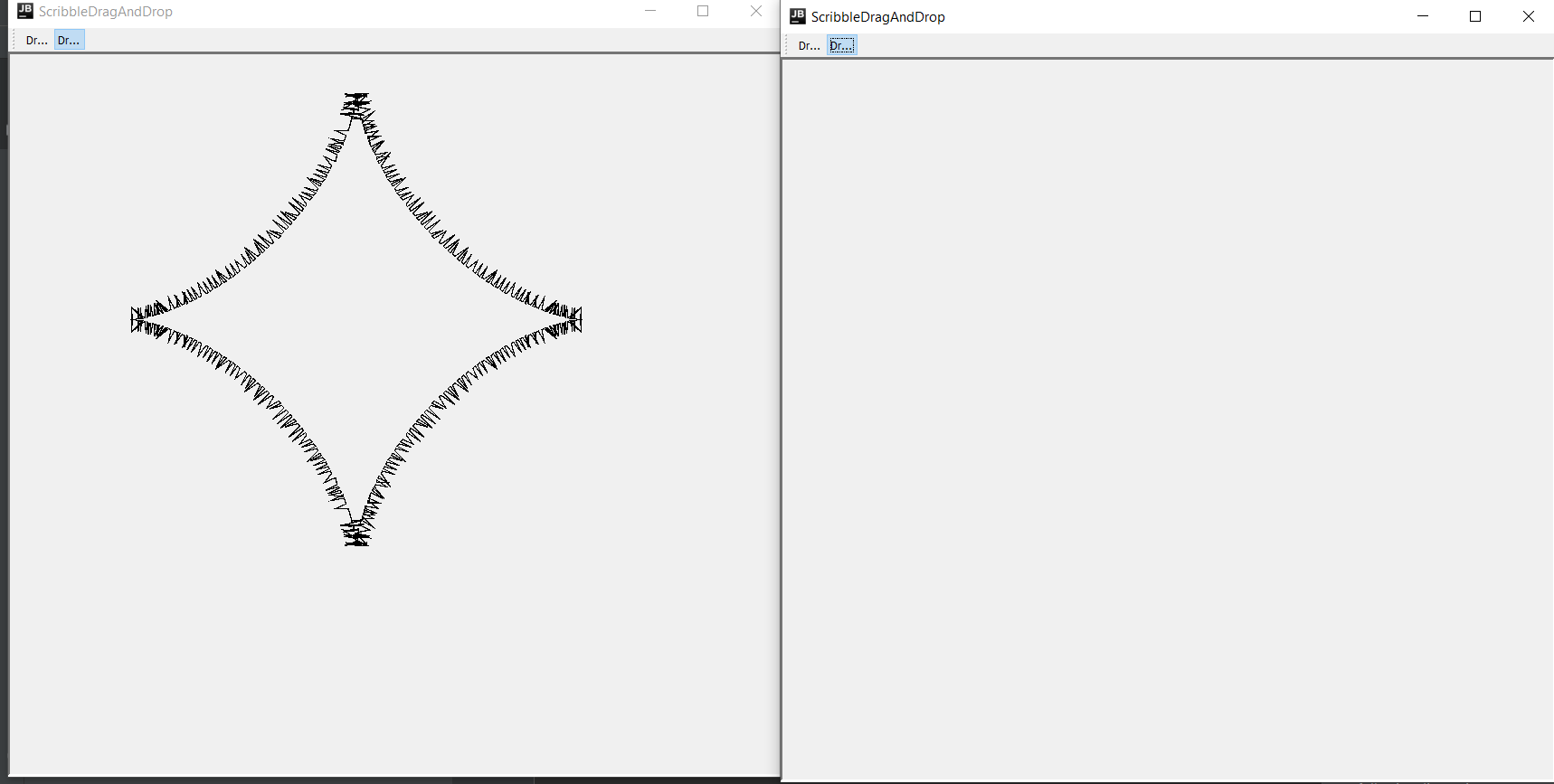
## Особенности реализации

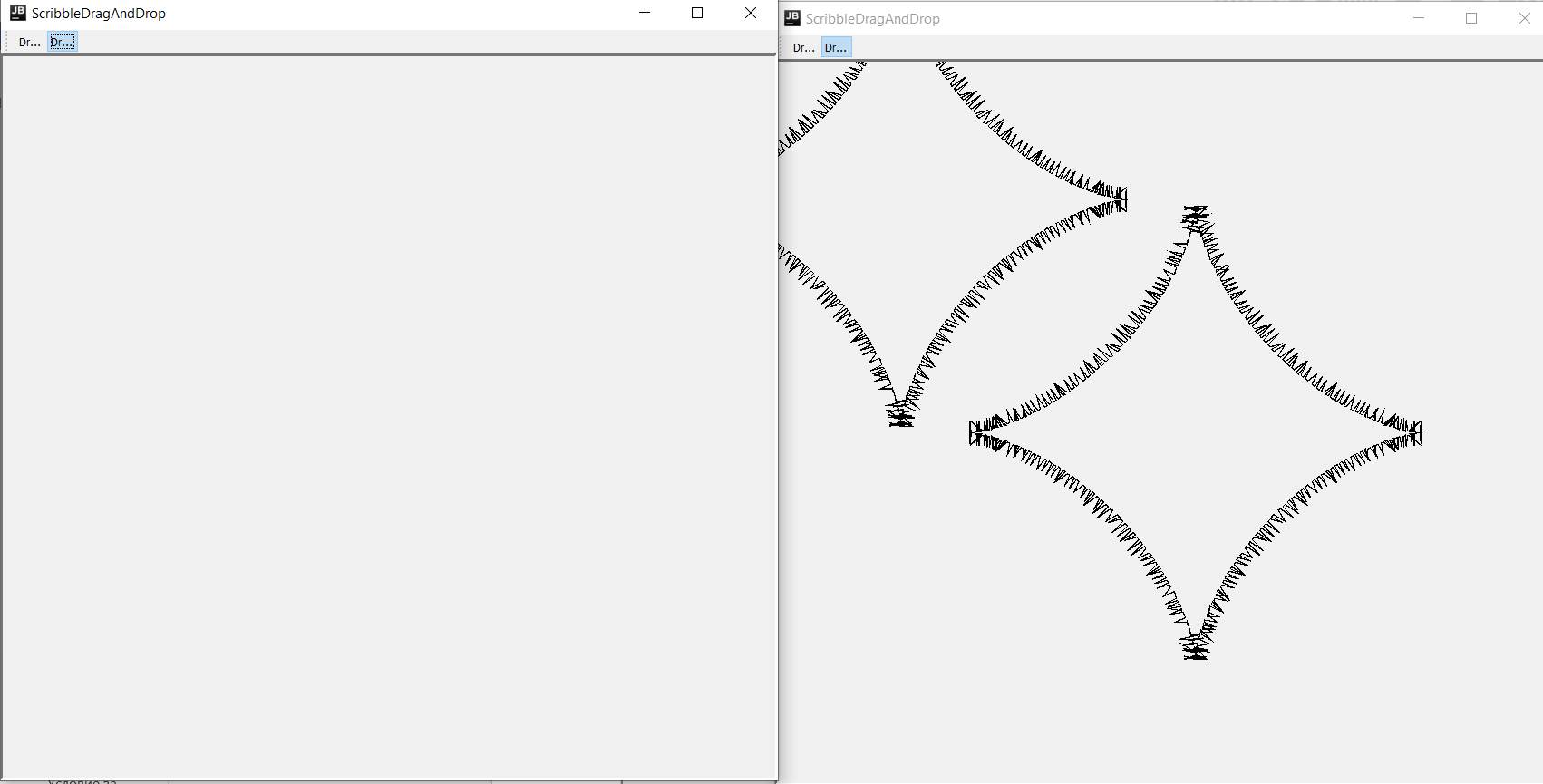
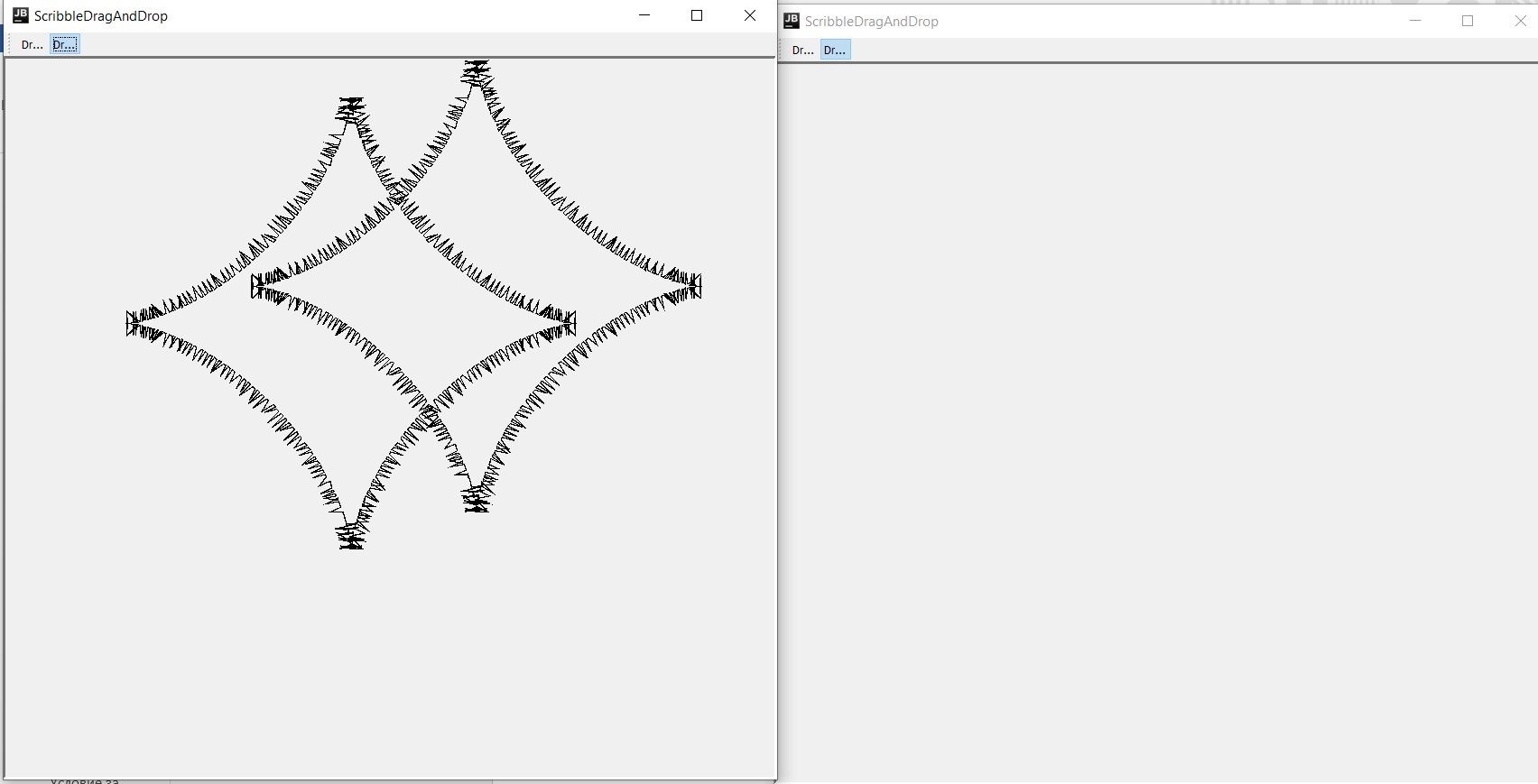
Под основу была взята работа из задания №3, реализованы интерфейсы:

Так же для клонировая рисунка был реализован интерфейс clonable



## Демонстрация работы программы





# Задание №7

## Условие задачи

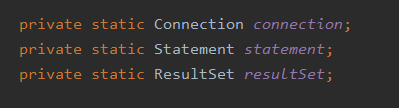
- Исследовать предложенную предметную область, спроектировать структуру базы данных объектов выбранной предметной области (из не менее чем 2-х таблиц объектов).

- 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

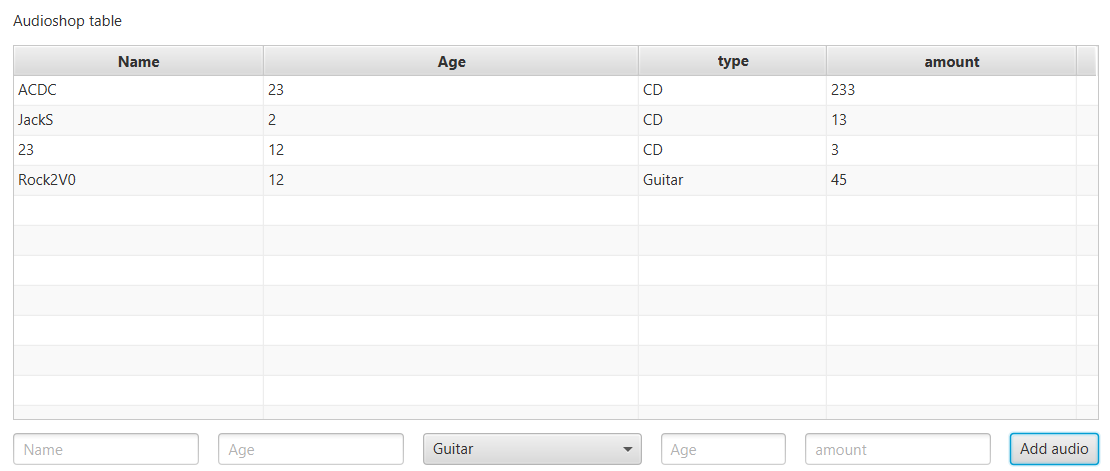
Работа идет с SQLlLite. Создали две таблицы для описания типа, и самих товаров. Работаю с БД при помощи следующих классов:

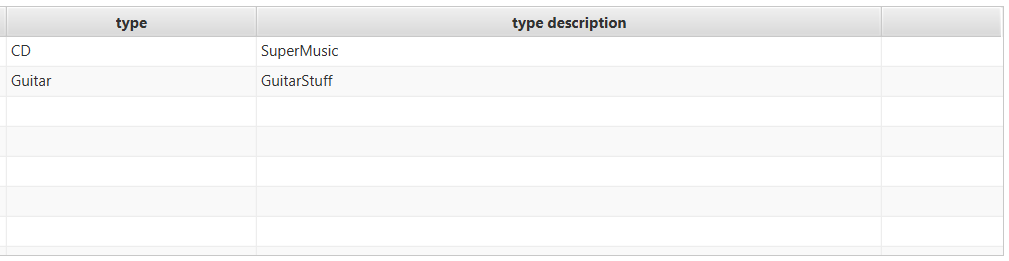


Фрагмент кода: подключение к БД



## Демонстрация работы программы





# 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

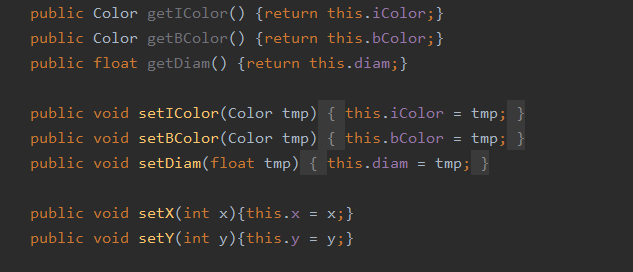
3) Создайте тестовое приложение в NetBeans с использованием вашего компонента.

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

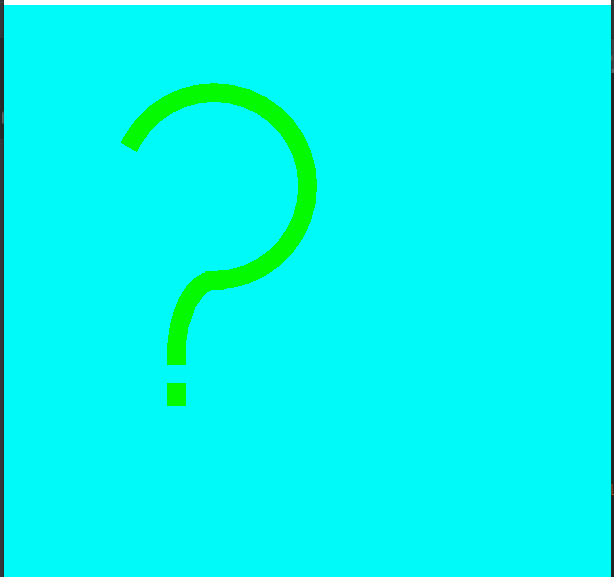
## Особенности реализации

За рисование отвечает класс Info, который унаследован от Canvas.

Данный класс предоставляет следующий интерфейс для редактирования своих свойств:



## 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.

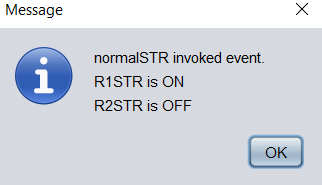
3) Создайте тестовое приложение в NetBeans с использованием вашего компонента.

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 AcceptEvent и интерфейс для подписчиков - AcceptListener. 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

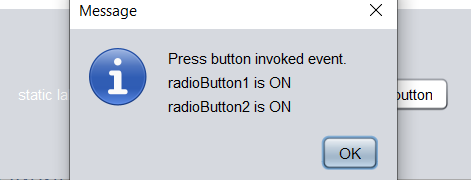
## Особенности реализации

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 возможные конфигурация соответвующих 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.

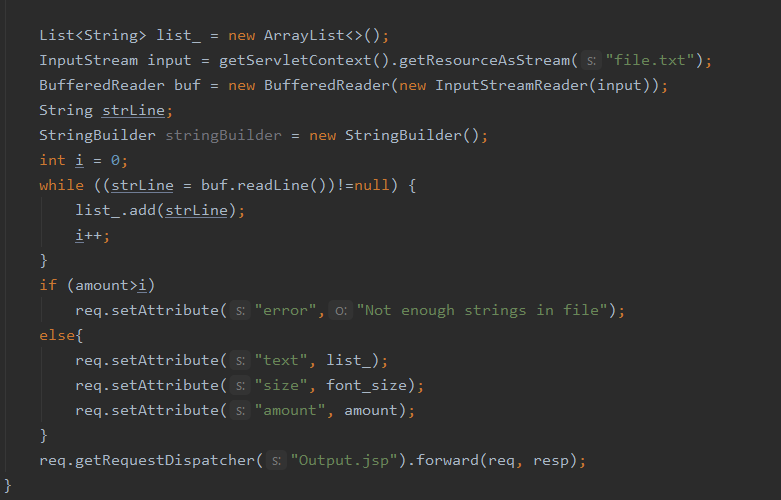
## Особенности реализации

Основной функция для свзяи между сервером и нашим сервелетом является функция doGet(). Данные метод вызывается всегда пока открыто окно в браузере и принмаетсә информацию о странице с сервера.

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

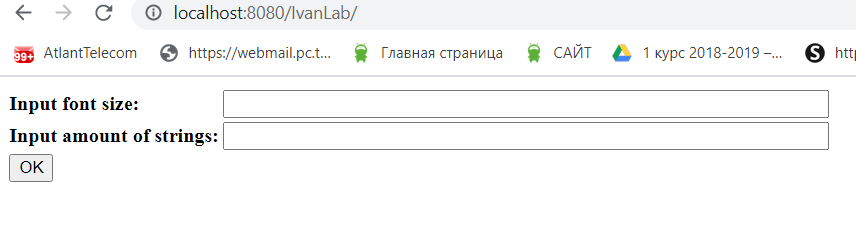
Текст для предоставления пользователю выбирался из заранее подготовленного текста. Использовался сервер tomcat 8.5.

Реализация предостваления текста:

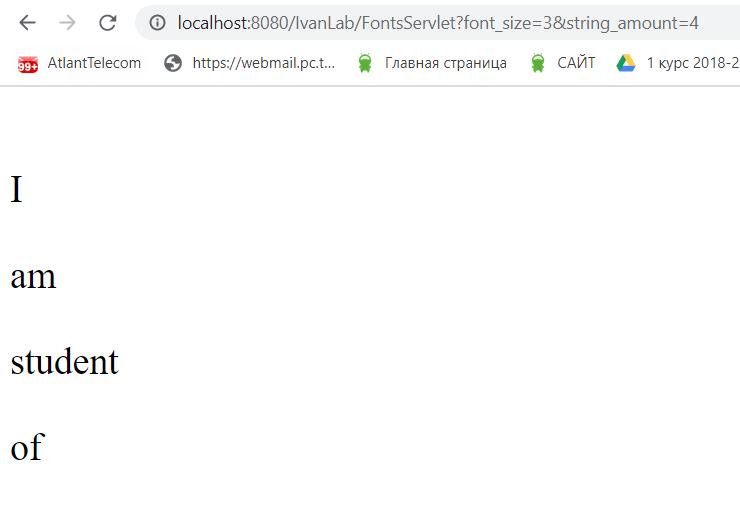


## Демонстрация работы программы

Запрос на необходимые параметры:



Вывод ответа на запрос



# 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.

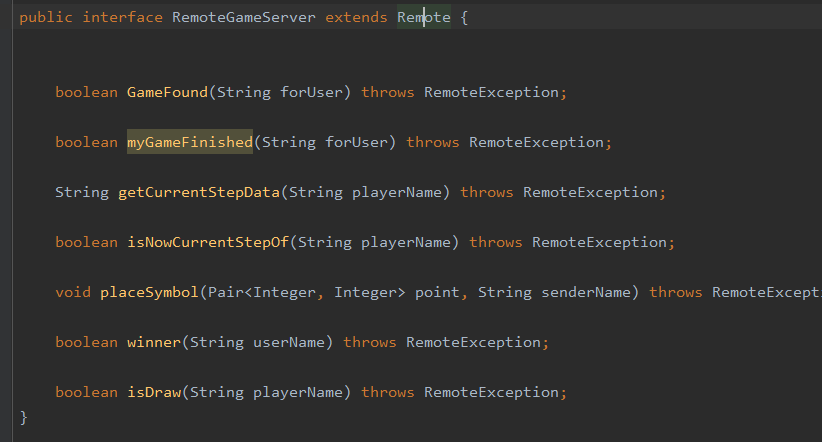
3) Создайте на основе технологии RMI клиент/серверное приложение:

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

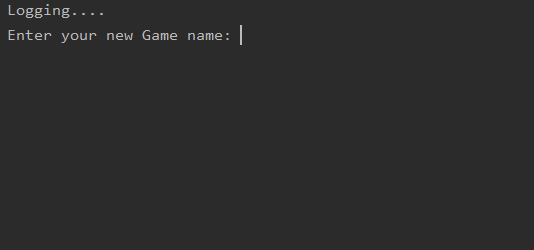
Основой в работе RMI является интерфейс, который наследуется от интерфейса Remote и необходим для определения функций которыми может пользоваться клиент на своей стороне.

В моем случае для игры я ограничился следующим интерфейсом

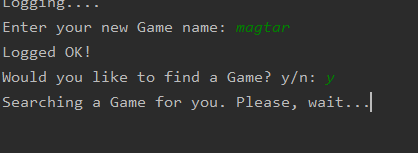


## Демонстрация работы программы

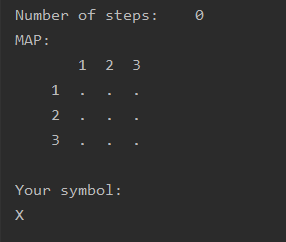
Подключение клиента:



Поиск игры



Начало игры



Конец игры

