БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Кафедра информационных систем управления

Специальность “Прикладная информатика”

**ОТЧЕТ**

по учебной практике

Студентка

группы 14Б, 2 курс \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е. А. Ермолаева

*(подпись)*

Руководитель,

ассистент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.В. Кудош

*(подпись)*

Минск

2021

**СОДЕРЖАНИЕ**

[**Задание №1**](#_vsvy4xu356dc) **2**

[**Задание №2**](#_ld27ovvhwvd4) **4**

[**Задание №3**](#_6hobdrat6gxh) **7**

[**Задание №4**](#_jvv5ra6l5q99) **9**

[**Задание №5**](#_112xdgn768b4) **12**

[**Задание №6**](#_86nbunpdant1) **14**

[**Задание №7**](#_e4dsyg328fan) **16**

[**Задание №8**](#_4evr05b3yek7) **18**

[**Задание №9**](#_z5voaa929vn7) **20**

[**Задание №10**](#_8uajnyu1iu1h) **22**

[**Задание №11**](#_krqg2hmkeixm) **24**

[**Задание №12**](#_s2q7rfnu00f3) **26**

# 

# Задание №1

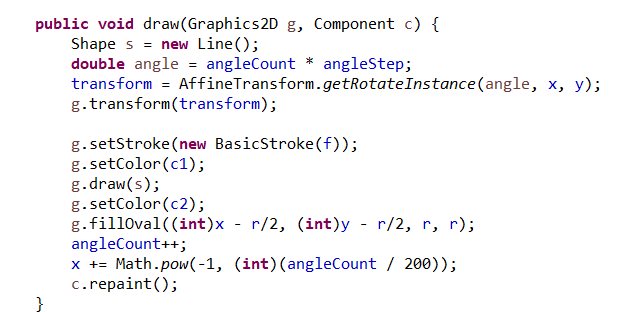
Постановка задачи:

* для изображения указанной в задании фигуры создать класс, реализующий интерфейс Shape
* выполнить указанные в задании перемещения указанной фигуры с помощью аффинного преобразования координат
* выполнить рисунок в окне апплета или фрейма с выбранной толщиной границы фигуры, цветом границы и цветом внутренней области (вводить толщину и цвет в качестве аргументов ваших программ или параметров апплета).

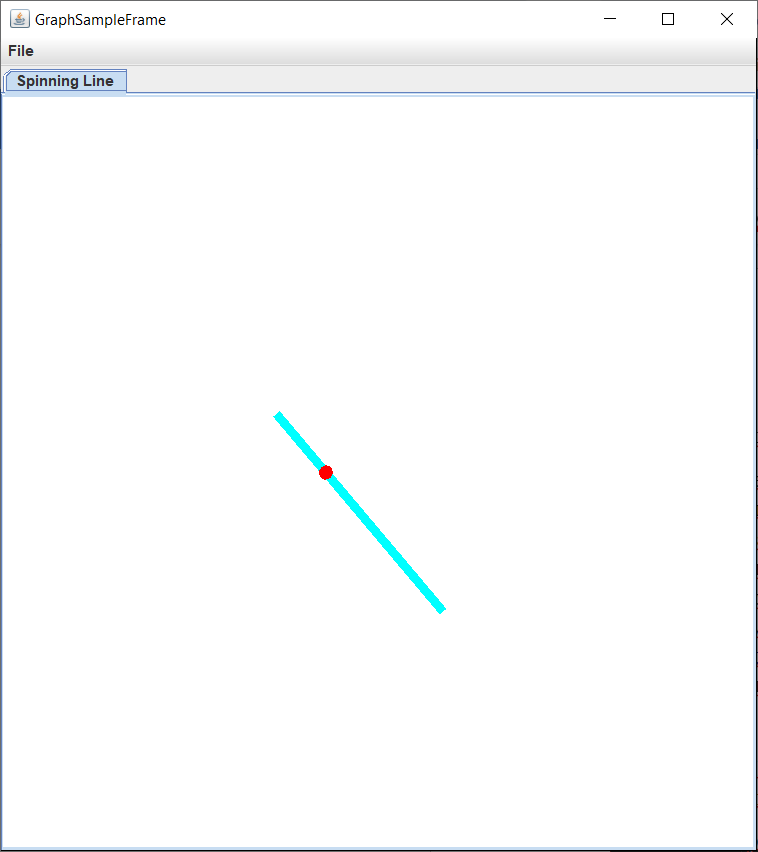
Вариант 4: Изобразить отрезок, вращающийся в плоскости экрана вокруг точки, движущейся по отрезку.

Особенности реализации:

Для изображения вращения отрезка используется аффинное преобразование:



Результат работы программы:



# Задание №2

Постановка задачи:

В следующих заданиях создайте тестовое приложение (Frame/JFrame) для демонстрации вашего решения, при этом:

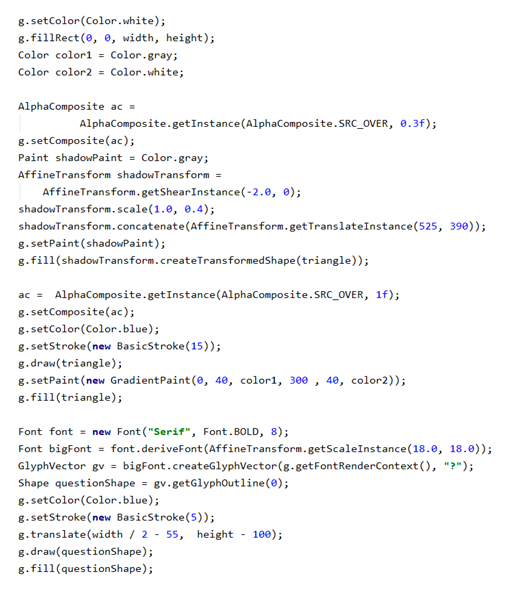
* для изображения указанной в задании фигуры создать класс, реализующий интерфейс Shape;
* создайте указанный фильтр изображения; при тестировании выведите фигуру без фильтра и с фильтром (аналогично фильтрам из примеров);
* моделируйте освещение и тень от объекта при помощи альфа-канала и/или механизма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

Вариант 16: Фигура (дорожный знак): вопросительный знак в равностороннем треугольнике, цвет треугольника и символа – синий, цвет фона – серый с градиентной заливкой слева-направо.

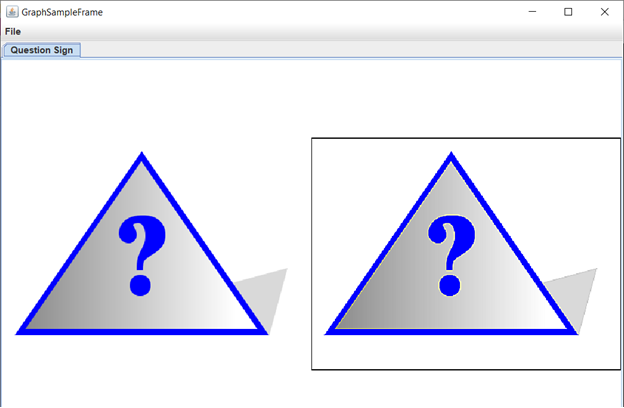
Фильтр: Sharpen

Особенности реализации:

Для изображения фигуры используется:



Результат работы программы:



# Задание №3

Постановка задачи:

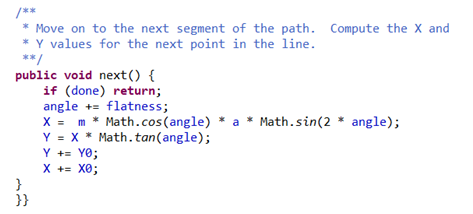
1. Разработайте пользовательский класс Shape реализующий рисование указанной алгебраической линии.
2. Разработайте пользовательский класс Stroke для изображения указанного контура, используя в качестве исходных точек результаты класса Shape, созданного на шаге 1)
3. Создайте приложение (Frame/JFrame)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

Вариант 7: Линия: четырехлепестковая роза

Контур: 

Особенности реализации:

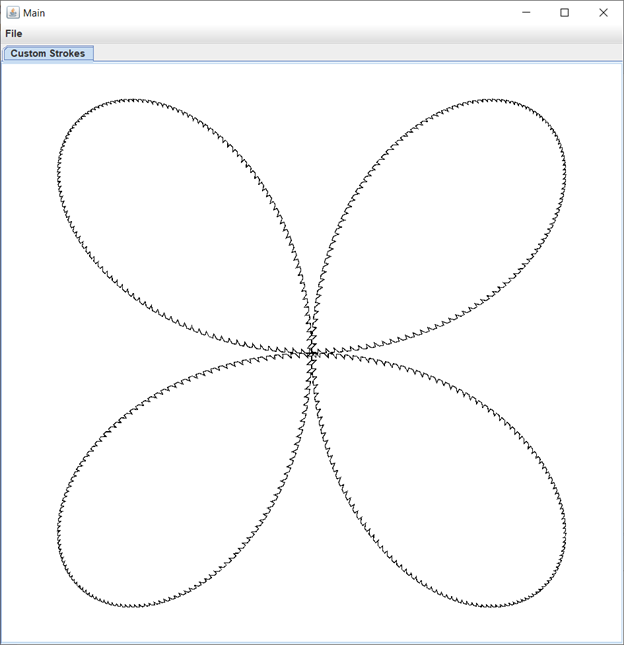
Для изображения алгебраической линии используется:



Для изображения контура используется:



Результат работы программы:



# 

# Задание №4

Постановка задачи:

Модифицируйте вашу программу следующим образом. В демонстрационное приложение добавьте возможность печати небольшого отчета о решении задания №3. Отчет должен содержать следующее:

* рисунок с подписью (по стандарту!) 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.

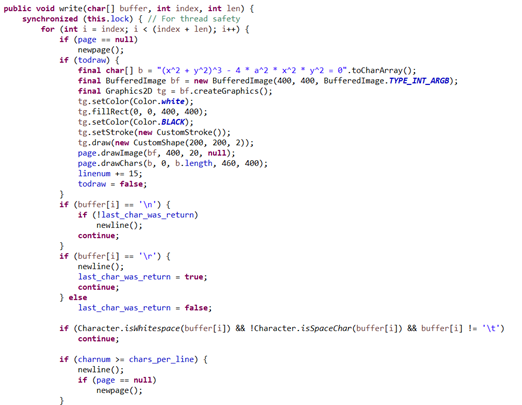
При печати используйте режим альбомной ориентации страницы и двустороннюю печать. Рисунок должен занимать не более половины страницы, при печати выровнять его по горизонтали.

Вариант 7: Линия: четырехлепестковая роза

Контур: 

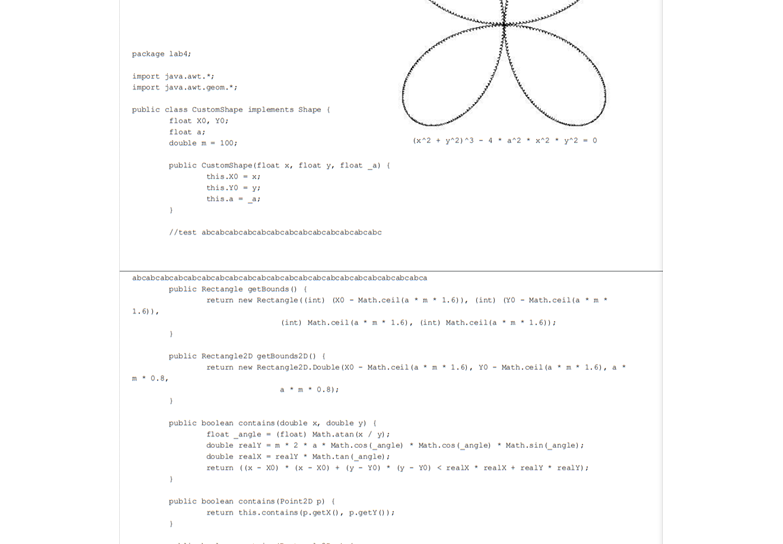
Особенности реализации:

Для печати отчета используется:





Результат работы программы:



# Задание №5

Постановка задачи:

1. Разработайте систему классов/интерфейсов для предметной области Вашего варианта задания. Данные необходимо упорядочить по атрибутам/свойствам товаров, предметов и т. п. в виде дерева.
2. 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)
3. Организуйте создание/загрузку/сохранение данных вашего варианта задания в файл (Вариант тестовых данных сдаётся вместе с исходным кодом задачи).

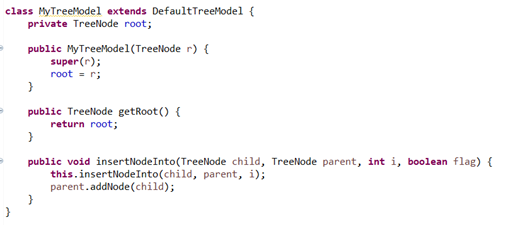
Вариант 10: Каталог магазина периферийных устройств компьютера (принтеры, сканеры, плоттеры, мониторы и т. п.)

Особенности реализации:

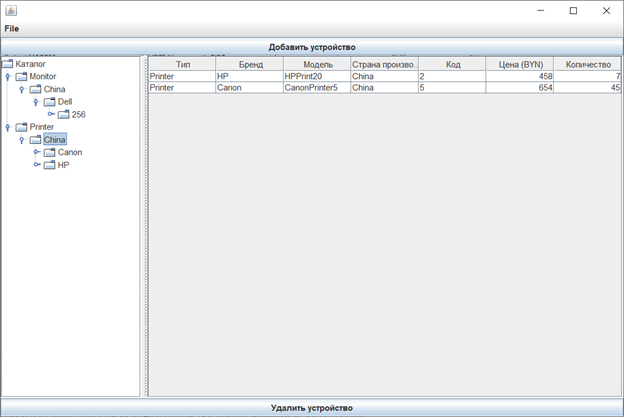
Для отображения таблицы используется:



Для отображения дерева используется:



Результат работы программы:



# Задание №6

Постановка 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.

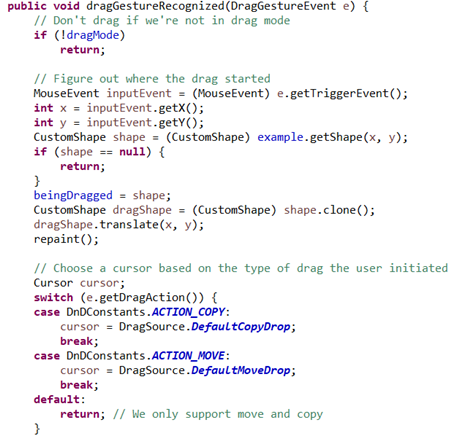
При реализации интерфейса тестового приложения следуйте рекомендациям стандарта CUI (Common User Interface).

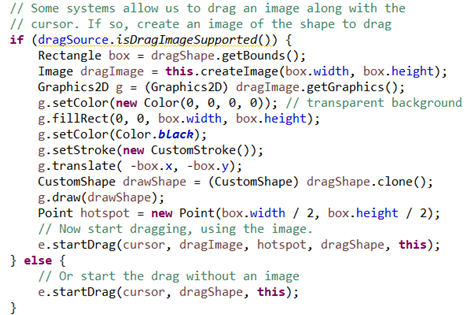
Вариант 7: Линия: четырехлепестковая роза

Контур: 

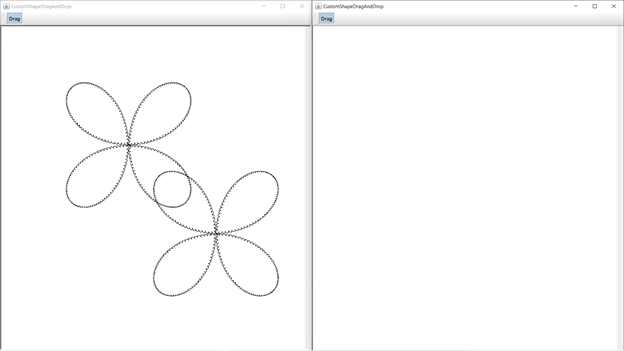
Особенности реализации:

Для реализации перетаскивания используется:





Результат работы программы:



# Задание №7

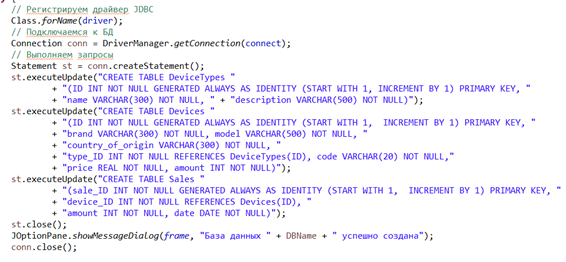
Постановка задачи:

1. Исследовать предложенную предметную область, спроектировать структуру базы данных объектов выбранной предметной области (из не менее чем 2-х таблиц объектов).
2. Разработайте графическое приложение для создания/ввода/отображения БД Вашего варианта задания. Содержимое БД отображайте в виде таблиц.
3. При реализации интерфейса следуйте рекомендациям стандарта CUI (Common User Interface).

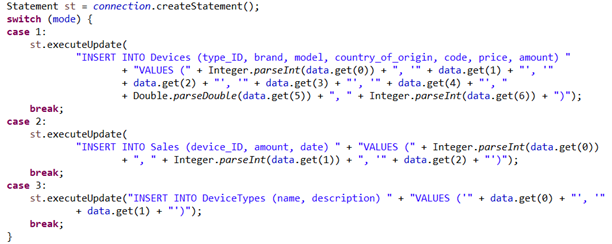
Вариант 10: Каталог магазина периферийных устройств компьютера (принтеры, сканеры, плоттеры, мониторы и т. п.)

Особенности реализации:

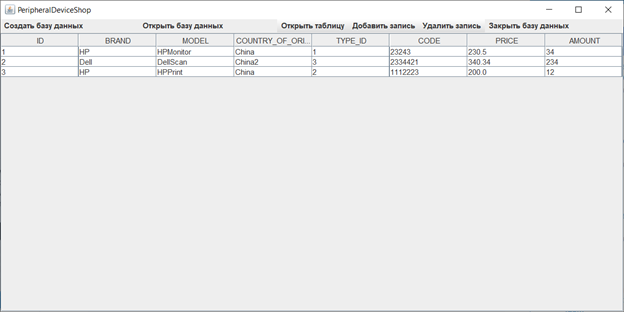
Для создания базы данных используется:



Для обновления базы данных используется:



Результат работы программы:



# Задание №8

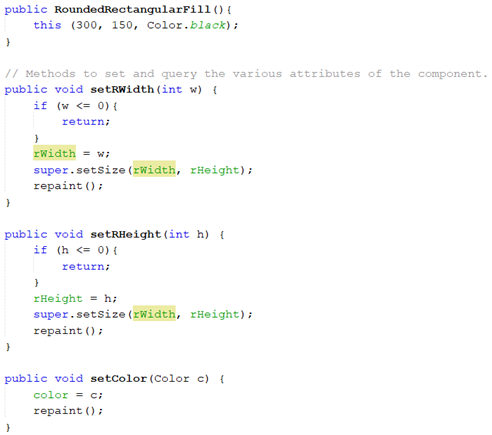
Постановка задачи:

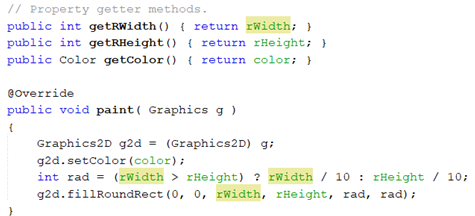
1. Изучите материал примера по быстрому введению в среду разработки NetBeans и компоненты JavaBeans по адресу: <http://docs.oracle.com/javase/tutorial/javabeans/quick/index.html>
2. Разработайте простой компонент вашего варианта задания на базе класса Canvas. Создайте файл манифеста и упакуйте компонент вместе с исходным кодом разработанных классов. При разработке поместите все ваши классы в пакет: bsu.fpmi.edupract
3. Создайте тестовое приложение в среде NetBeans с использованием вашего компонента.

Вариант 7: Прямоугольная заливка с скругленными углами. Свойства: ширина, высота и цвет.

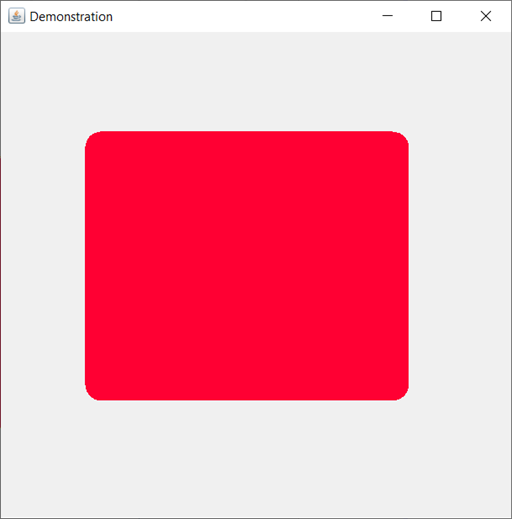
Особенности реализации:

Для реализации компонента используется:





Результат работы программы:



# Задание №9

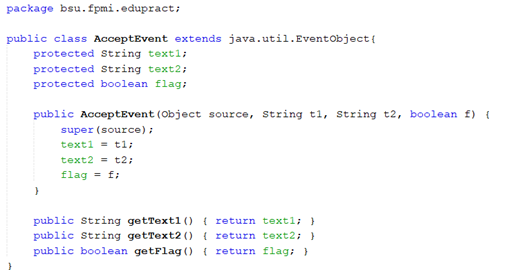
Постановка задачи:

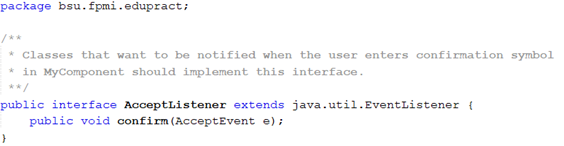
1. Разработайте компонент вашего варианта задания. Создайте файл манифеста и упакуйте компонент вместе с исходным кодом разработанных классов. При разработке поместите все ваши классы в пакет: bsu.fpmi.edupract
2. Компонент должен реализовывать класс BeanInfo с информацией о компоненте.
3. Создайте тестовое приложение в NetBeans с использованием вашего компонента.

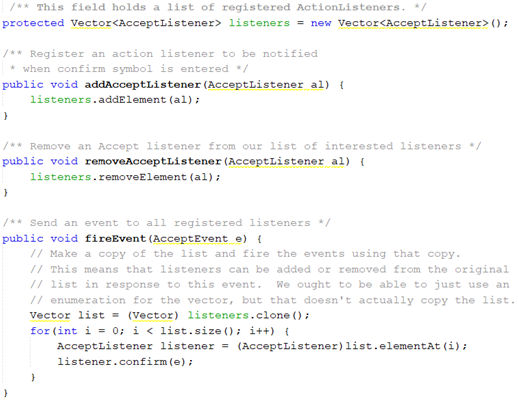
Вариант 15: Однострочный статический текст, две строки ввода и кнопка-флаг. Свойства: текст, текст флага, символ подтверждения. Событие генерируется при вводе символа подтверждения. Событие передает еще состояние флага и содержимое строк ввода.

Особенности реализации:

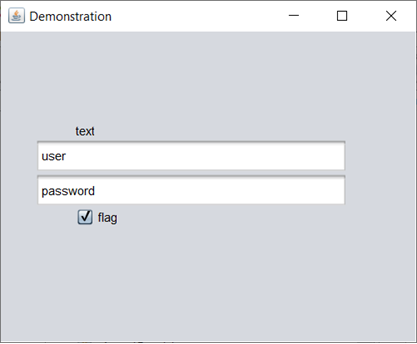
Для реализации события используется:







Результат работы программы:



# Задание №10

Постановка задачи:

Для решения задания №10 используем решенный вариант задания №9. Номера заданий сохраняются. Модифицируем тип свойства компонента так, как указано ниже;

*Основная задача:*

Создаем собственный редактор для каждого свойства компонента. Каждый редактор ограничивает возможные значения свойства, предоставляя выбор из списка трех-пяти допустимых значений (т. е. определяем методы getTags()).

Регистрируем редакторы в классе BeanInfo компонента.

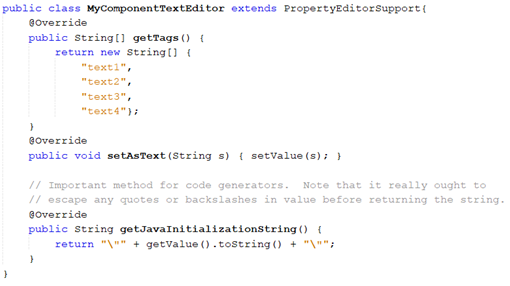
*Дополнительно:*

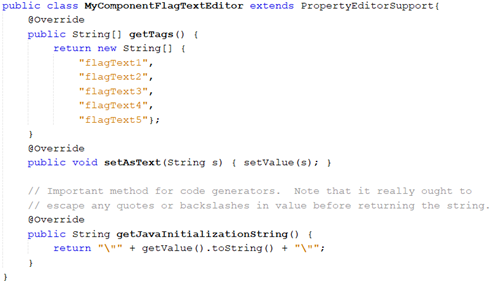
Попытайтесь создать настройщик компонента, который позволит менять списки допустимых значений для свойств вашего компонента.

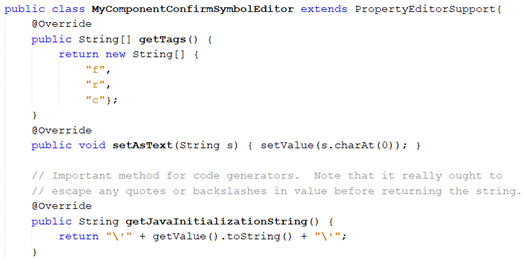
Вариант 15: Однострочный статический текст, две строки ввода и кнопка-флаг. Свойства: текст, текст флага, символ подтверждения. Событие генерируется при вводе символа подтверждения. Событие передает еще состояние флага и содержимое строк ввода.

Особенности реализации:

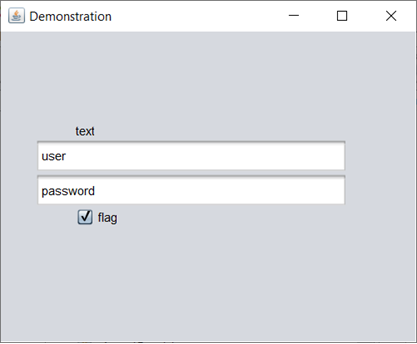
Для реализации редакторов свойств используется:







Результат работы программы:



# Задание №11

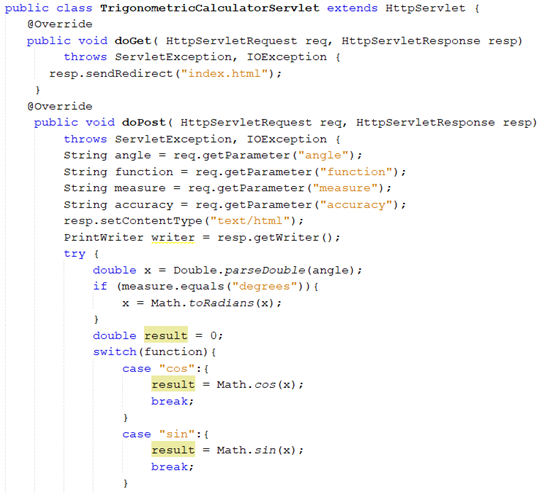
Постановка задачи:

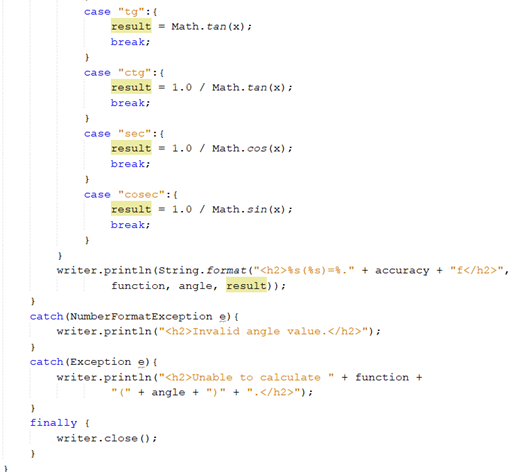
Создать сервлет и взаимодействующие с ним пакеты Java-классов и HTML-документов, выполняющие действия для решения вашего варианта задания. Представить решение в виде web-приложения (как в примере).

Вариант 2: Вычисление тригонометрических функций в градусах и радианах с указанной точностью. Выбор функции должен осуществляться через выпадающий список.

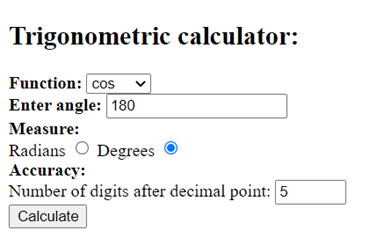
Особенности реализации:

Для реализации сервлета используется:





Результат работы программы:





# Задание №12

Постановка задачи:

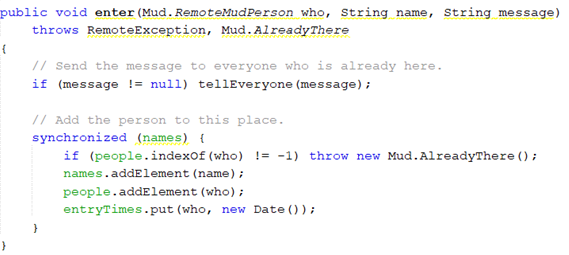
1. Изучите пример 2.
2. Проанализируйте ваш вариант задания. Можно ли его реализовать как часть MUD системы (например, в одной из комнат MudPlace), требуется ли для этого внести изменения в парадигму MUD? Какие изменения потребует реализация клиента MUD, другие классы примера? Оформите эти размышления в вашем отчете 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:

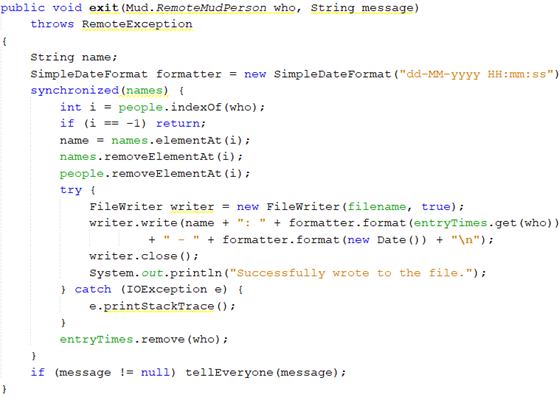
Вариант 7: Учет рабочего времени. Сервер ведет учет времени работы клиентов, данные сохраняются в файле. Клиент при запуске связывается с сервером и сообщает данные клиента. Сервер каждые 5 мин запрашивает подтверждение у клиента, что он еще подключен. Если клиент не отвечает, он закончил работу.

Особенности реализации:

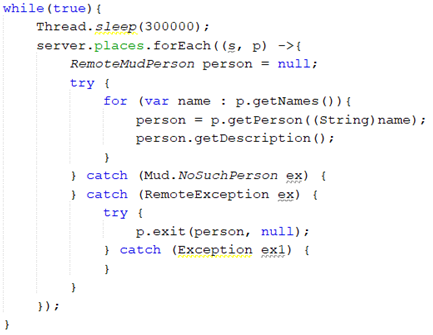
Вариант задания можно реализовать как часть MUD системы. Для этого не понадобится вносить изменения в парадигму MUD. Достаточно будет лишь внести следующие изменения в классы примера:

MudPlace:





MudServer:



Результат работы программы:

