

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**  
**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  
**Факультет прикладной математики и информатики**

Жуковский Павел Сергеевич  
**Отчёт по учебной практике**  
студента 2 курса 13 группы

**Руководитель практики**  
Орешко Игорь Георгиевич

Минск 2020

## Оглавление

Лабораторная работа №1 .....	3
Лабораторная работа №2 .....	4
Лабораторная работа №3 .....	6
Лабораторная работа №4 .....	7
Лабораторная работа №5 .....	9
Лабораторная работа №6 .....	11
Лабораторная работа №7 .....	12
Лабораторная работа №8 .....	13
Лабораторная работа №9 .....	14
Лабораторная работа №10 .....	16
Лабораторная работа №11 .....	17
Лабораторная работа №12 .....	20
Список использованной литературы .....	21

## Лабораторная работа №1

### Постановка задачи:

- 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).

### Вариант 5

#### Особенности реализации:

Изобразить четырехугольник, вращающийся в плоскости экрана вокруг своего центра тяжести.

Результат работы программы (Рисунок 1):

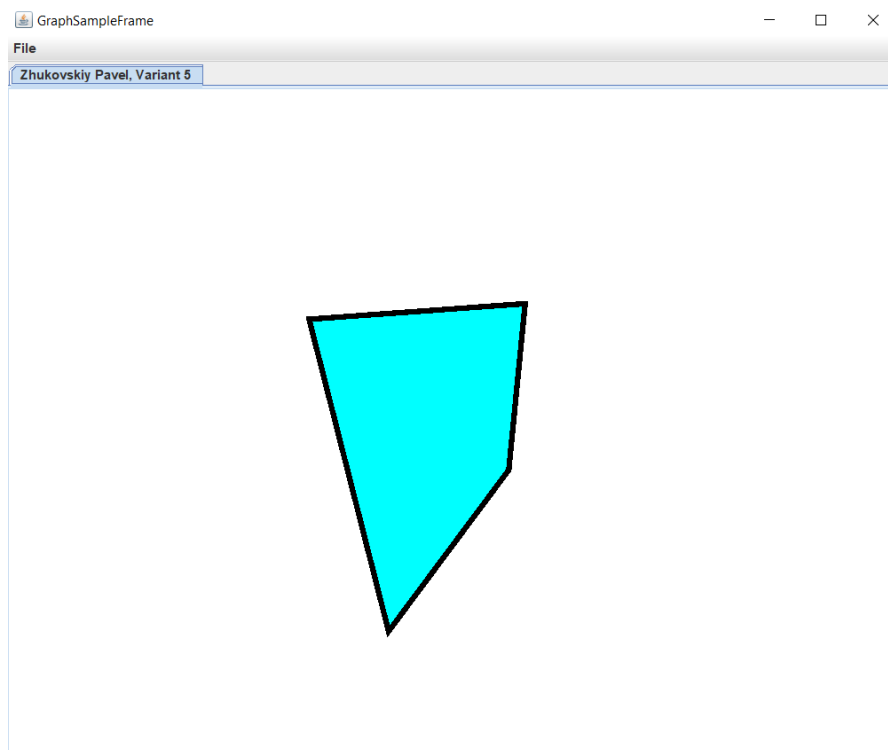


Рисунок 1

## **Лабораторная работа №2**

### Постановка задачи:

- Для изображения указанной в задании фигуры создать класс, реализующий интерфейс Shape;
- Создать указанный фильтр изображения; при тестировании вывести фигуру без фильтра и с фильтром (аналогично фильтрам из примеров);
- Моделировать освещение и тень от объекта при помощи альфа-канала и/или механизма обработки изображения;
- При рисовании использовать сглаживание, внеэкранный буфер и преобразования координат.

### **Вариант 5**

#### Особенности реализации:

Фигура (дорожный знак): надпись WAIT в прямоугольнике, цвет прямоугольника и надписи – жёлтый, цвет фона – серый с градиентной заливкой снизу-вверх

Фильтр: Rotate CW 45 degrees (поворот по часовой на 45)

#### Результат работы программы (Рисунок 2):



Рисунок 2

## Лабораторная работа №3

### Постановка задачи:

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

### Вариант 14

#### Особенности реализации:

Линия: Обыкновенная циклоида

$$x = a \cos^{-1} \left( 1 - \frac{y}{a} \right) - \sqrt{2ay - y^2}$$

Контур: 

#### Результат работы программы (Рисунок 3):

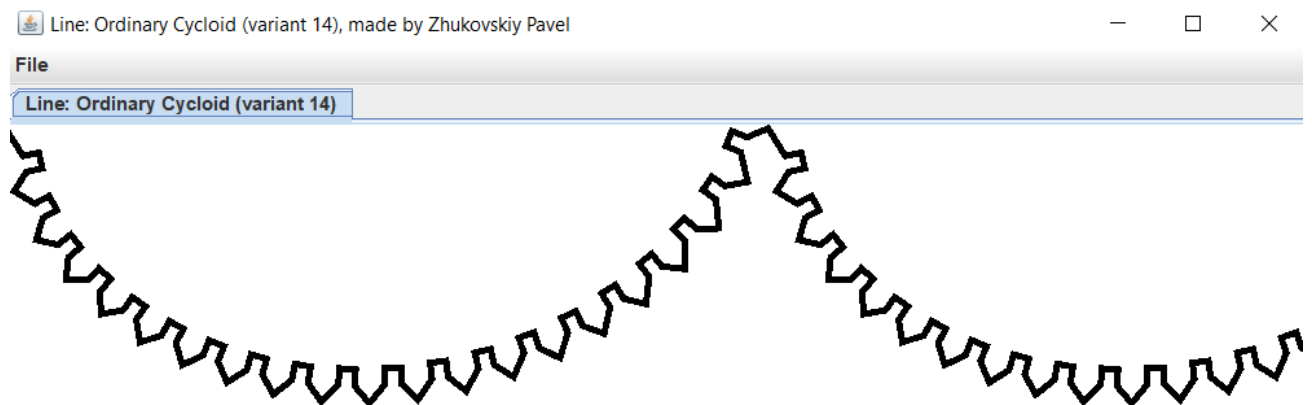


Рисунок 3

## Лабораторная работа №4

### Постановка задачи:

Модифицировать программу следующим образом. В демонстрационное приложение добавить возможность печати небольшого отчёта о решении задания No3. Отчёт должен содержать следующее:

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

Для длинных строк, выходящих за границы области печати, организовать перенос текста на новую строку с разрывом по пробельным символам. При печати использовать режим альбомной ориентации страницы и двустороннюю печать. Рисунок должен занимать не более половины страницы, при печати выравнивать его по горизонтали.

### Вариант 14

#### Особенности реализации:

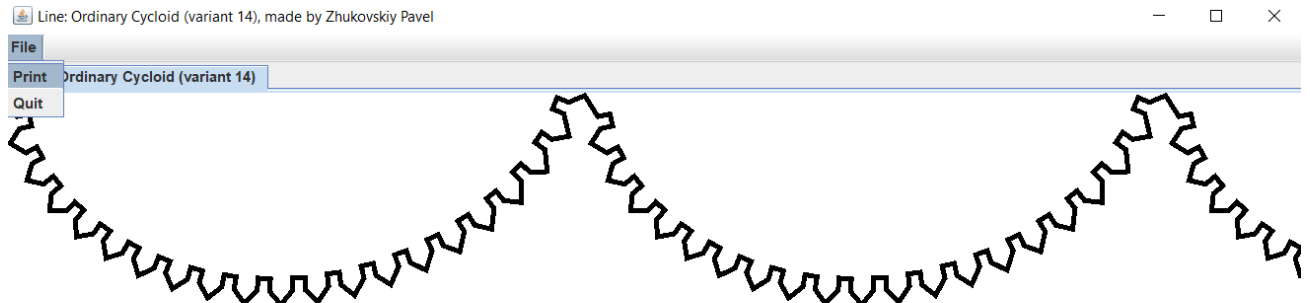
Линия: Обыкновенная циклоида

$$x = a \cos^{-1} \left( 1 - \frac{y}{a} \right) - \sqrt{2ay - y^2}$$

Контур:



#### Результат работы программы (Рисунок 4):



Source code:

```

import java.awt.Rectangle;
import java.awt.Shape;
import java.awt.geom.AffineTransform;
import java.awt.geom.PathIterator;
import java.awt.geom.Point2D;
import java.awt.geom.Rectangle2D;

public class MyShape implements Shape {
    private int centerX;
    private int centerY;
    private int param;

    MyShape(int centerX, int centerY, int param) {
        this.centerX = centerX;
        this.centerY = centerY;
        this.param = param;
    }

    @Override
    public Rectangle getBounds() {
        return null;
    }

    @Override
    public Rectangle2D getBounds2D() {
        return null;
    }

    @Override
    public boolean contains(double x, double y) {
        return false;
    }

    @Override
    public boolean contains(Point2D p) {
        return false;
    }

    @Override
    public boolean intersects(double x, double y, double w, double h) {
        return false;
    }

    @Override
    public boolean intersects(Rectangle2D r) {
        return false;
    }

    @Override
    public boolean contains(double x, double y, double w, double h) {
        return false;
    }

    @Override
    public boolean contains(Rectangle2D r) {
        return false;
    }

    @Override
    public PathIterator getPathIterator(AffineTransform at) {
        return new ShapeIterator();
    }

    class ShapeIterator implements PathIterator {
        boolean done = false;
        double h = Math.PI / 500;
        boolean start = true;
        double t = -2;

        @Override
        public int getWindingRule() {
            return 0;
        }

        @Override
        public boolean isDone() {
            return done;
        }

        @Override
        public void next() {
            t += h;
        }
    }

    public int currentSegment(float[] coordinate) {
        if (start) {
            coordinate[0] = (float) (param + t - param * Math.sin(t)); // x
            coordinate[1] = (float) (param - param * Math.cos(t)); // y
            start = false;
            return SEG_MOVETO;
        }
        if (t >= 10 * Math.PI) {
            done = true;
            return SEG_CLOSE;
        }
        coordinate[0] = (float) (param + t - param * Math.sin(t)); // x
        coordinate[1] = (float) (param - param * Math.cos(t)); // y
        return SEG_LINETO;
    }

    @Override
    public int currentSegment(double[] coordinate) {
        if (start) {
            coordinate[0] = (double) (param + t - param * Math.sin(t)); // x
            coordinate[1] = (double) (param - param * Math.cos(t)); // y
            start = false;
            return SEG_MOVETO;
        }
        if (t >= 10 * Math.PI) {
            done = true;
            return SEG_CLOSE;
        }
        coordinate[0] = (double) (param + t - param * Math.sin(t)); // x
        coordinate[1] = (double) (param - param * Math.cos(t)); // y
        return SEG_LINETO;
    }
}

```

Рисунок 4



## **Лабораторная работа №5**

### Постановка задачи:

- Разработать систему классов/интерфейсов для предметной области Вашего варианта задания. Данные необходимо упорядочить по атрибутам/свойствам товаров, предметов и т.п. в виде дерева.
- Разработать графическое приложение для ввода/отображения данных Вашего варианта задания. При отображении структуры данных в виде дерева реализовывать интерфейс `javax.swing.Tree.TreeModel`. Листья дерева отображать в виде таблицы, для этого реализовывать интерфейс `javax.swing.table.TableModel`. (пример похожего приложения – Проводник Windows).
- При реализации интерфейса следовать рекомендациям стандарта CUI (Common User Interface).

## **Вариант 10**

### Особенности реализации:

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

### Результат работы программы (Рисунок 5):

	Количество	Название
peripherals	7 штук	Экранный ГП Huion Kamvas GT-156HD v2 15.6"
GraphicsTablets	14 штук	Экранный ГП WACOM CINTIQ 13HD CREATIVE PEN DISPLAY DTK1300
ScreenGraphicsTablets	23 штуки	Безэкранный ГП Wacom (CTL-471)
Экранный ГП 1	11 штук	Безэкранный ГП HUION 420
Экранный ГП 2	43 штуки	Проводные наушники Rdtmx RH-529
UnscreenGraphicsTablets	51 штука	Проводные наушники Sennheiser HD 280 Pro
Безэкранный ГП 1	72 штуки	Проводные наушники Panasonic RP-HT161
Безэкранный ГП 2	29 штук	Проводные наушники Sony MDR-EX650
Headphones	19 штук	Проводные наушники Shure SE215
WiredHeadphones	84 штуки	Беспроводные наушники Apple AirPods Pro
Проводные наушники 1	78 штук	Беспроводные наушники HBDQ-68
Проводные наушники 2	89 штук	Механическая Клавиатура Logitech G413 (черная)
Проводные наушники 3	72 штуки	Механическая Клавиатура Harper GKB-P100
Проводные наушники 4	65 штук	Механическая клавиатура Gembird KB-G540L
Проводные наушники 5	48 штук	Механическая клавиатура Corsair K95 RGB Platinum SE (Cherry MX Speed)
WirelessHeadphones	71 штука	Мембранная клавиатура Logitech G213 Prodigy [920-008092]
Беспроводные наушники 1	41 штука	Мембранная клавиатура Logitech K120
Беспроводные наушники 2	94 штуки	Мембранная клавиатура HyperX Alloy Core RGB
Keyboards	23 штуки	Беспроводная клавиатура Logitech Wireless Touch Keyboard K400 Plus Black (920-0
WiredKeyboards	18 штук	Беспроводная клавиатура Defender UltraMlate SM-535 RU / 45535 (черная)
MechanicalKeyboards	14 штук	Проводной Микрофон Audio-Technica AT2020
Механическая клавиатура 1	17 штук	Проводной Микрофон Marshall Electronics MXL 770
Механическая клавиатура 2	29 штук	Проводной Микрофон eE Electronics X1
Механическая клавиатура 3	31 штука	Проводной Микрофон BEHRINGER C-1U
Механическая клавиатура 4	50 штук	Беспроводной Микрофон Magic Karaoke YS-68 (чёрный)
MembraneKeyboards	97 штук	Проводная Мышь Logitech G102 Prodigy Gaming Mouse Black USB
Мембранная клавиатура 1	85 штук	Проводная Мышь ASUS UT200 Black USB
Мембранная клавиатура 2	98 штук	Проводная Мышь HP X500 Wired Mouse E5E76AA Black USB
Мембранная клавиатура 3	45 штук	Беспроводная мышь Logitech M170 / 910-004042
Мембранная клавиатура 4	41 штука	Беспроводная мышь Logitech M280 / 910-004287
WirelessKeyboards		
Беспроводная клавиатура 1		
Беспроводная клавиатура 2		
Microphones		
WiredMicrophones		
Проводной микрофон 1		
Проводной микрофон 2		
Проводной микрофон 3		
Проводной микрофон 4		
WirelessMicrophones		
Беспроводной микрофон 1		
Mouses		
WiredMouses		
Проводная мышь 1		
Проводная мышь 2		

Рисунок 5

## Лабораторная работа №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 14

### Особенности реализации:

Линия: Обыкновенная циклоида

$$x = a \cos^{-1} \left( 1 - \frac{y}{a} \right) - \sqrt{2ay - y^2}$$

### Результат работы программы (Рисунок 6):

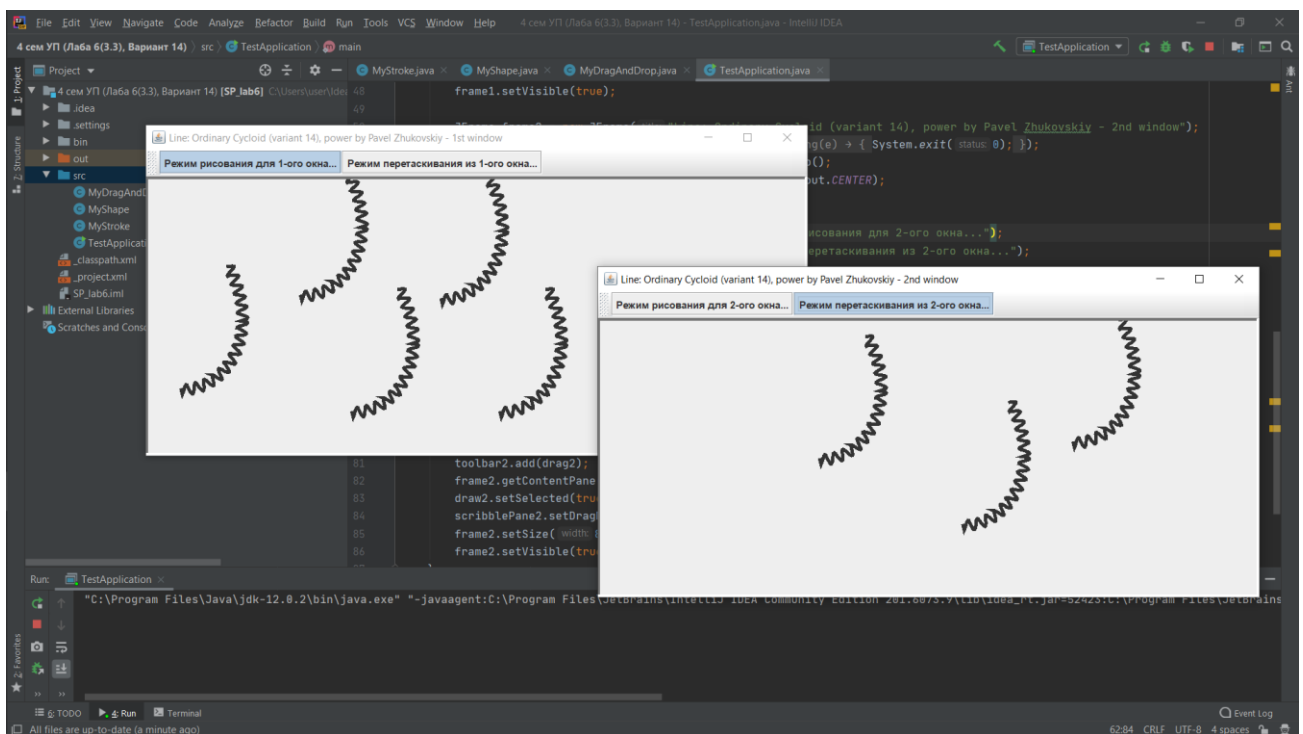


Рисунок 6

## Лабораторная работа №7

### Постановка задачи:

Исследовать предложенную предметную область, спроектировать структуру базы данных объектов выбранной предметной области (из не менее чем 2-х таблиц объектов). Согласовать проект БД с преподавателем. Обязательно работать с Derby. Разработать графическое приложение для создания/ввода/отображения БД варианта задания. Содержимое БД отображать в виде таблиц. При реализации интерфейса следовать рекомендациям стандарта CUI (Common User Interface).

### Вариант 10

#### Особенности реализации:

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

#### Результат работы программы (Рисунок 7):

```
public class Peripheral
{ // Класс "Периферийные устройства"
    final String Name;
    final int ID_Manufact_City;
    final int ID_Manufact_Street;
    final int Year_of_creation;
    final int ID_Supplier;
}
```

Город производителя	Улица производства	Год создания	Поставщик
инск	Макаёнка	2016	Даниил Лебедев
инск	Октябрьская	2019	Андрей Гришин
инск	Макаёнка	2019	Андрей Гришин
инск	Макаёнка	2019	Андрей Гришин
аршава	Киселёва	2020	Павел Жуковский
озырь	Октябрьская	2019	Павел Жуковский
инск	Дзержинского	2020	Павел Жуковский
инск	Дзержинского	2020	Павел Жуковский
инск	Дзержинского	2019	Павел Жуковский
инск	Дзержинского	2020	Павел Жуковский
инск	Беломорская	2020	Павел Жуковский
инск	Максима Богдановича	2019	Павел Жуковский
инск	Машерова	2019	Павел Жуковский
инск	Машерова	2020	Павел Жуковский
инск	Кирова	2020	Павел Жуковский
инск	Кирова	2020	Павел Жуковский
инск	Ленина	2020	Павел Жуковский
инск	Чкалова	2020	Павел Жуковский
инск	Сурганова	2018	Павел Жуковский
инск	Орловская	2019	Павел Жуковский
инск	Филимонова	2019	Павел Жуковский
инск	Козлова	2018	Павел Жуковский
инск	Уральская	2017	Павел Жуковский
Минск	Тростенечная	2020	Павел Жуковский
Минск	Есенина	2020	Павел Жуковский
Берлин	Независимости	2018	Анна Гридишко
Берлин	Независимости	2017	Анна Гридишко
Киев	Громова	2019	Илья Гузей
Воронеж	Яна Чечота	2019	Егор Шилкин
Воронеж	Яна Чечота	2016	Егор Шилкин
Гомель	Сурацкая	2015	Илья Салей
Гродно	Жукова	2014	Виктор Грасыков
Могилев	Пуликова	2017	Дмитрий Тарасенко
Могилев	Пуликова	2020	Дмитрий Тарасенко
Витебск	Толстого	2018	Анастасия Кацуба
Варшава	Киселёва	2019	Александра Малякю
Беласток	Платонова	2013	Константин Корткевич
Владивосток	Петруса Бровки	2018	Глеб Леоцкий
Нью-Йорк	Беломорская	2020	Елизавета Мойсейчик

Рисунок 6

## Лабораторная работа №8

### Постановка задачи:

- 1) Изучить 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) Разработать простой компонент вашего варианта задания на базе класса Canvas. Создать файл манифеста и упаковать компонент вместе с исходным кодом разработанных классов. При разработке поместить все классы в пакет: bsu.fpmi.educational\_practice
- 3) Создать тестовое приложение в NetBeans с использованием компонента.

### Вариант 5

### Особенности реализации:

- 5) 3D прямоугольник. Свойства: ширина и высота

### Результат работы программы (Рисунок 8):

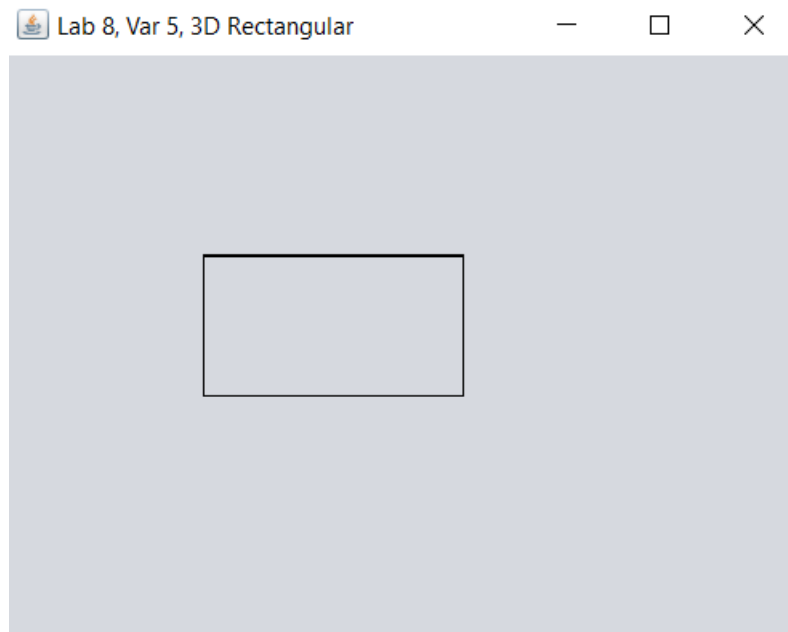


Рисунок 7

## **Лабораторная работа №9**

### **Постановка задачи:**

1) Разработать компонент варианта задания. Создать файл манифеста и упаковать компонент вместе с исходным кодом разработанных классов. При разработке поместить все классы в пакет:

bsu.fpmi.educational\_practice

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

3) Создать тестовое приложение в NetBeans с использованием компонента.

Интерфейсные компоненты с реализацией собственного события AcceptEvent.

Определить также интерфейс слушателя вашего события AcceptListener:

Передавать слушателю события информацию о том, в результате чего произошло событие, если событие может генерироваться от нескольких действий пользователя.

### **Вариант 3**

#### **Особенности реализации:**

3) Однострочный статический текст, две зависимых радио-кнопки и обычная кнопка. Свойства: текст, текст кнопки, текст радио-кнопок. Событие генерируется при нажатии на обычную кнопку. Событие передаёт ещё и состояние радио-кнопок.

#### **Результат работы программы (Рисунок 9):**

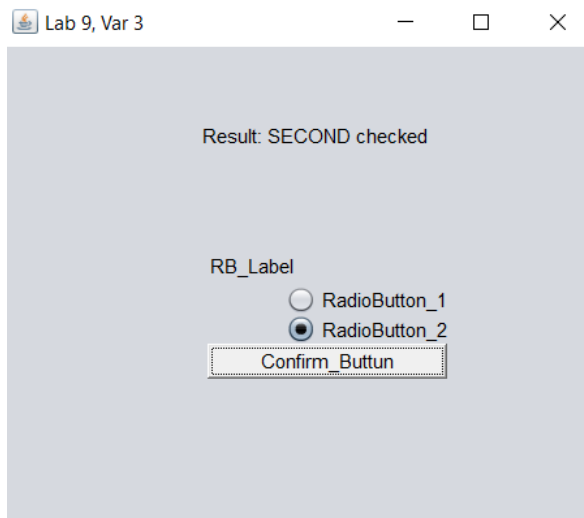


Рисунок 8

## Лабораторная работа №10

### Постановка задачи:

Для решения задания №10 использовать решённый вариант задания №9. Номера заданий сохраняются. Модифицировать тип свойства компонента так, как указано ниже:

Создать собственный редактор для каждого свойства компонента. Каждый редактор ограничивает возможные значения свойства, предоставляя выбор из списка трёх – пяти допустимых значений (т. е. определить методы `getTags()`). Регистрировать редакторы в классе `BeanInfo` компонента.

### Вариант 3

#### Особенности реализации:

3) Однострочный статический текст, две зависимых радио-кнопки и обычная кнопка. Свойства: текст, текст кнопки, текст радио-кнопок. Событие генерируется при нажатии на обычную кнопку. Событие передаёт ещё и состояние радио-кнопок.

#### Результат работы программы (Рисунок 10):

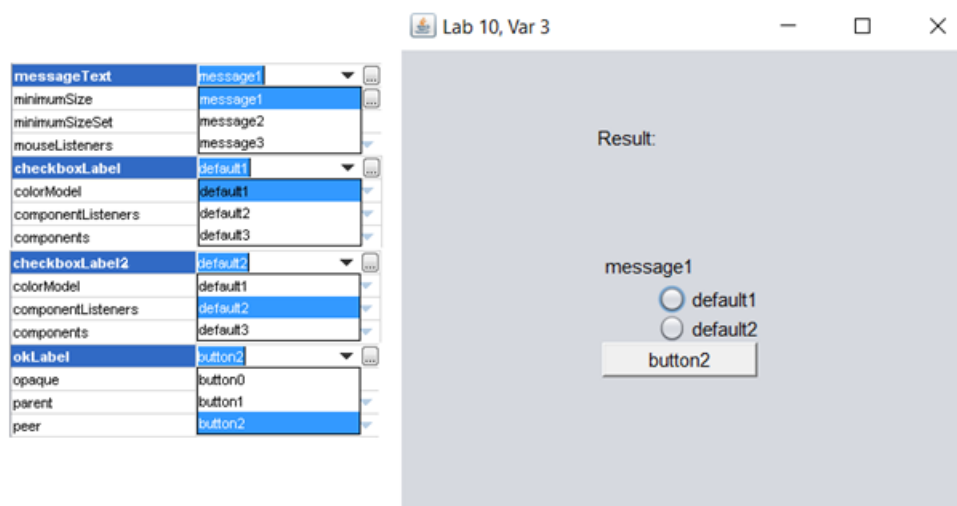


Рисунок 9



## Лабораторная работа №11

### Постановка задачи:

Создать сервлет и взаимодействующие с ним пакеты Java-классов и HTML-документов, выполняющие действия для решения варианта задания.

Представить решение в виде web-приложения (как в примере).

### Вариант 9

### Особенности реализации:

9) Вычисление арифметического выражения из действительных чисел и знаков 4-х арифметических операций.

### Результат работы программы (Рисунки 11, 12, 13):

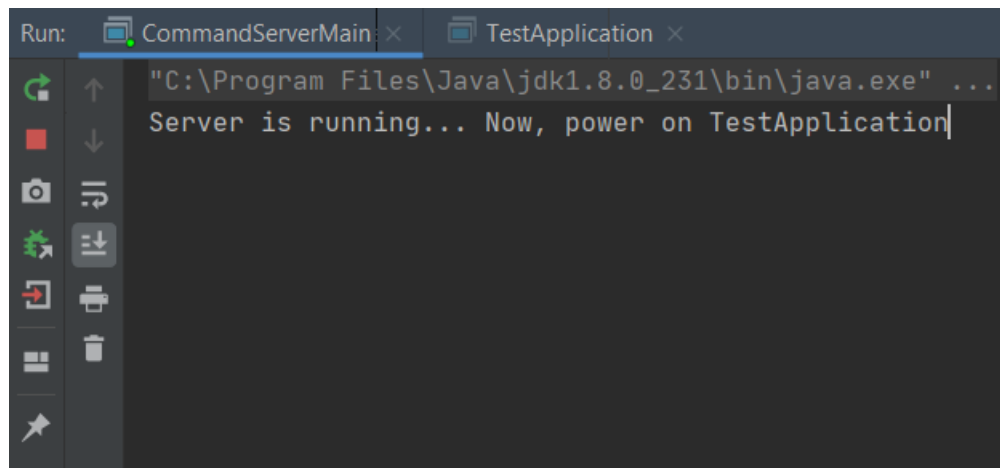
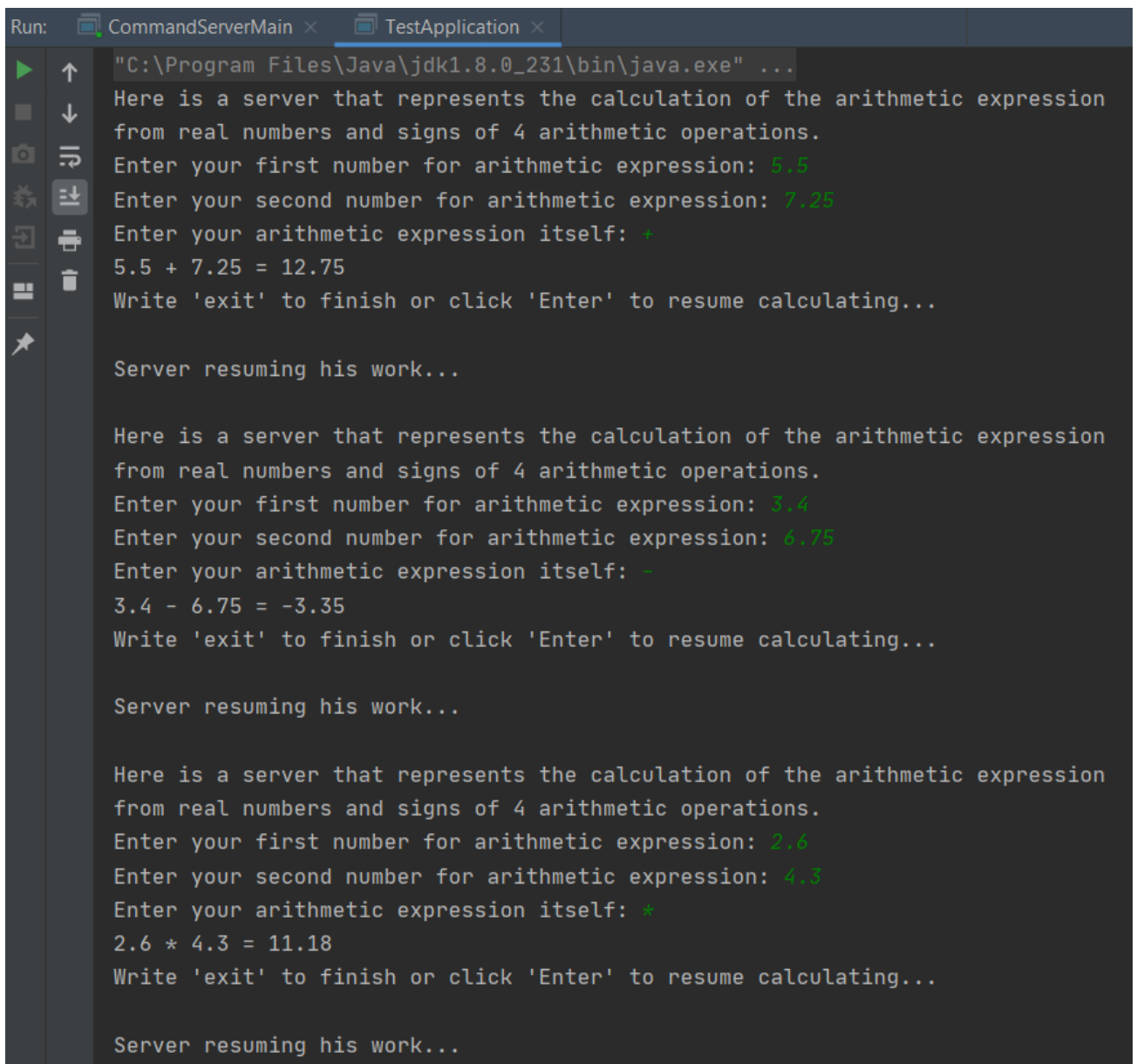


Рисунок 10



The screenshot shows a Java IDE with two tabs: "CommandServerMain" and "TestApplication". The "TestApplication" tab is active, displaying a command window with the following text:

```
"C:\Program Files\Java\jdk1.8.0_231\bin\java.exe" ...  
Here is a server that represents the calculation of the arithmetic expression  
from real numbers and signs of 4 arithmetic operations.  
Enter your first number for arithmetic expression: 5.5  
Enter your second number for arithmetic expression: 7.25  
Enter your arithmetic expression itself: +  
5.5 + 7.25 = 12.75  
Write 'exit' to finish or click 'Enter' to resume calculating...  
  
Server resuming his work...  
  
Here is a server that represents the calculation of the arithmetic expression  
from real numbers and signs of 4 arithmetic operations.  
Enter your first number for arithmetic expression: 3.4  
Enter your second number for arithmetic expression: 6.75  
Enter your arithmetic expression itself: -  
3.4 - 6.75 = -3.35  
Write 'exit' to finish or click 'Enter' to resume calculating...  
  
Server resuming his work...  
  
Here is a server that represents the calculation of the arithmetic expression  
from real numbers and signs of 4 arithmetic operations.  
Enter your first number for arithmetic expression: 2.6  
Enter your second number for arithmetic expression: 4.3  
Enter your arithmetic expression itself: *  
2.6 * 4.3 = 11.18  
Write 'exit' to finish or click 'Enter' to resume calculating...  
  
Server resuming his work...
```

Рисунок 11

Server resuming his work...

Here is a server that represents the calculation of the arithmetic expression from real numbers and signs of 4 arithmetic operations.

Enter your first number for arithmetic expression: 12.6

Enter your second number for arithmetic expression: 4.2

Enter your arithmetic expression itself: /

12.6 / 4.2 = 3.0

Write 'exit' to finish or click 'Enter' to resume calculating...

Server resuming his work...

Here is a server that represents the calculation of the arithmetic expression from real numbers and signs of 4 arithmetic operations.

Enter your first number for arithmetic expression: 55.94

Enter your second number for arithmetic expression: 0

Enter your arithmetic expression itself: /

java.lang.Exception: Division by zero is impossible!!!

at TestApplication.main([TestApplication.java:47](#))

Process finished with exit code 0

Рисунок 13

## Лабораторная работа №12

### 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.

### Вариант 8

#### Особенности реализации:

8) Трансляция новостей. Сервер хранит новостные сообщения за месяц. Все подключённые клиенты получают текущие новостные сообщения. Клиент также может запросить распечатку новостей за указанный день.

#### Результат работы программы (Рисунок 14):

На компьютер был установлен контейнер сервлетов Tomcat. На серверной стороне Controller обрабатывает запросы и отправляет ответы на страницу .jsp. На сервере все вопросы и рейтинги хранятся в специальных файлах. Добавить на сервер новость не представляет особых трудностей, как и получить информацию о них пользователями:

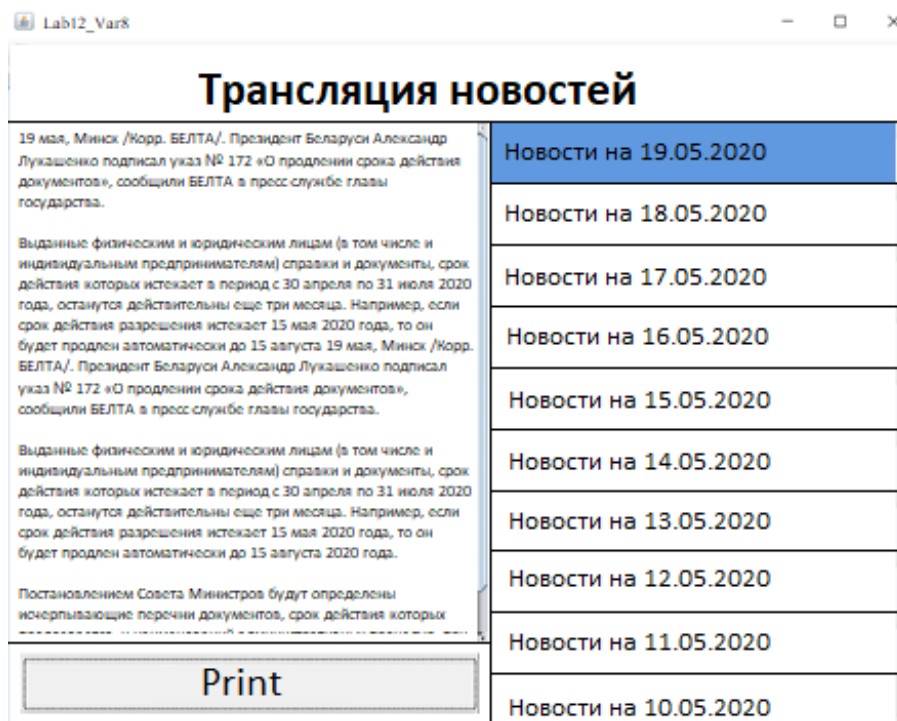


Рисунок 14

## **Список использованной литературы**

1. Хабибуллин И. Ш. Java 7. — СПб.: БХВ-Петербург, 2012 г.
2. Г. Шилдт. Java . Полное руководство, 8-е издание, 2012 г.
3. Кей С. Хорстман. Java2 Основы. Том 1. С.-Петербург., 2006 г.
4. Кей С. Хорстман. Java2 Тонкости программирования. Том 2. Петербург. 2007 г.