# Оглавление

[Оглавление 1](#_Toc509537085)

[Условие задачи 2](#_Toc509537086)

[Цель программы 2](#_Toc509537087)

[Исходные данные 2](#_Toc509537088)

[Выходные данные 2](#_Toc509537089)

[Математическая модель, являющаяся основой алгоритма 2](#_Toc509537090)

[Контрольные пример 2](#_Toc509537091)

[Условие задачи 2](#_Toc509537092)

[Решение 2](#_Toc509537093)

[Описание алгоритма в псевдокодах 4](#_Toc509537094)

[Блок-схема 4](#_Toc509537095)

[Описание структуры программы на Java 6](#_Toc509537096)

[Элементы интерфейса и порядок работы с ним (на примере java-программы) 8](#_Toc509537097)

[Основные функции программы 8](#_Toc509537098)

[UML диаграмма классов 11](#_Toc509537099)

[Код программы 12](#_Toc509537100)

[Класс Run 12](#_Toc509537101)

[Класс StarrPage 13](#_Toc509537102)

[Класс MainFrame 17](#_Toc509537103)

[Класс About 20](#_Toc509537104)

[Класс Autor 22](#_Toc509537105)

[Список использованных источников 26](#_Toc509537106)

# Условие задачи

1. Создать программу для нахождения обьема и площади шара.

* Для примера могут быть использованы любые целые положительные числа
* В программе предусмотреть ввод численного значения и нахождение величин с визуализацией результата.
* Для ввода и вывода данных используются соответствующие элементы управления
* Предусмотрена обработка ошибок (исключений) при неправильном вводе данных или других действиях пользователя.
* Программа позволяет выполнять все перечисленные действия с помощью необходимых элементов управления.
* Выбор элементов управления должен соответствует их назначению.
* Программа сдержит четыре окна (формы): стартовую, основную, сведения об авторе, сведения о программе.
* Программа разработана на языке Java.

# Цель программы

Цель программы – найти площадь и обьем шара при помощи введенного пользователем числа с визуализацией результата. Дополнительными функциями программы являются возможности очищать результат.

# Исходные данные

Основными исходными данными являются:

1. Исходное число, которое принимается за радиус шара.

# Выходные данные

1. Результат в виде площади шара
2. Результат в виде обьема шара

# Математическая модель, являющаяся основой алгоритма

Для нахождения значения объема и площади шара возможно использовать следующие формулы:

Где V – обьем шара в м^3,

S – площадь шара в м^2,

R – введенный пользователем радиус шара в м.

# Контрольные пример

### Условие задачи

Есть число 5. Найти площадь и обьем шара

### Решение

Формула обьема

4/3 \* 3.14 \* 5^3 = 523.59

Формула площади

4 \* 3.14 \* 5^2 = 314,159

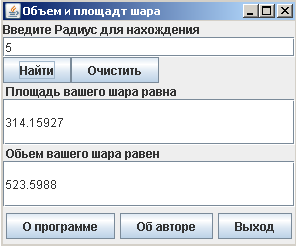


Рисунок 6.1 – Результат решения контрольного примера с использованием разработанной java-программы

# Описание алгоритма в псевдокодах

1. Начало, переход на 2
2. Поиск картинки заднего фона. При нахождении перейти на 3. При отсутствии вывести сообщение ошибки и перейти на 3
3. Открытие стртовой формы и ожидание выбора пользователя, в случае выбора «Далее» переход на 4, в случае «Выход» закрытие программы – переход к пункту 11
4. Водится Радиус и осуществляется переход к пункту 5.
5. Опрос событий нажатия кнопки «Найти». Если нажали, проверка на число: если не число, вывести сообщение и переход к 4, если число – переход к 6
6. Найти площадь и обьем, заполнить необходимые поля, переход к 7
7. Если нажата кнопка «Очистить» - очистить все текстовые поля и переход к 4, иначе переход к 8
8. Если нажата кнопка «Об авторе» - открыть новое окно «Об авторе», иначе переход к 9
9. Если нажата кнопка «О программе» - открыть новое окно «О программе», если нет – переход к 10
10. Если нажата кнопка «Выход» - переход к 11, если нет – переход к 4
11. Конец

# Блок-схема

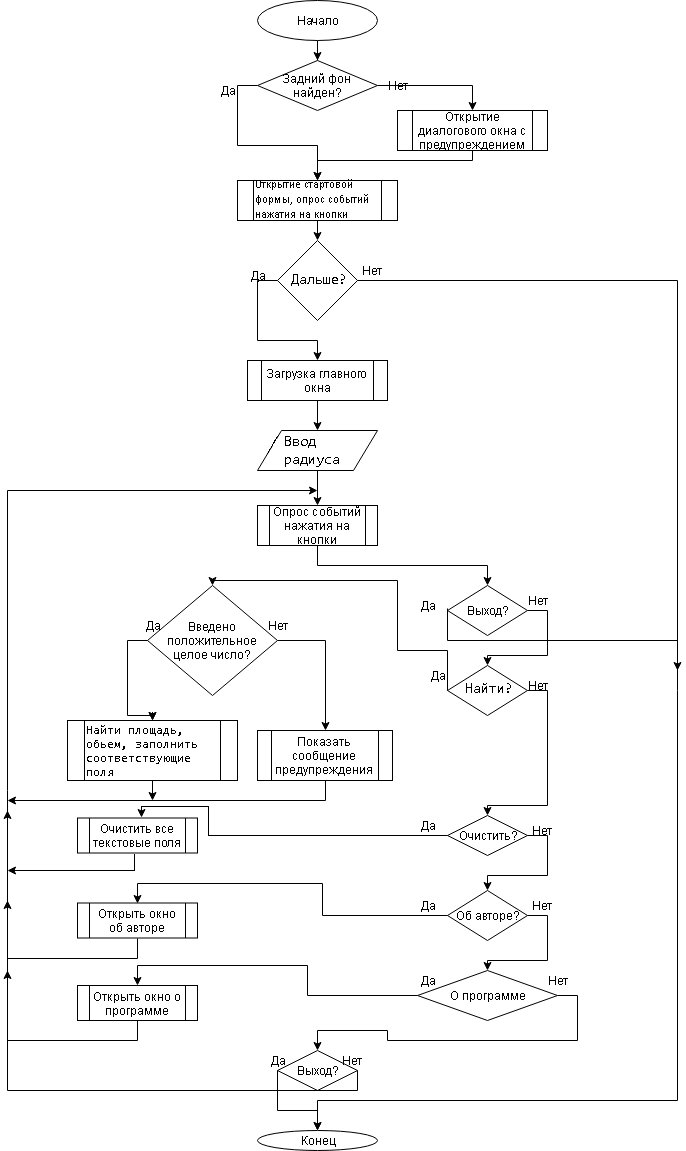


Рисунок 8.1 – Блок-схема алгоритма

# Описание структуры программы на Java

Java-программа состоит из следующих классов:

**MainFrame**- Нахождение площади и обьема. Основное окно.

**Run** - Запускающий класс (главный класс).

**StartPage** - Окно заставка.

**About** - Окно «О программе».

**Avtor** - Окно «Об авторе».

**BgPanel** – Класс отрисовки фона

**MainFrame –**. В данном классе создается основная форма для работы с полльзователем.

. Содержит поля

* JButton jbtCalc;
* JButton jbtClear;
* JButton jbtAbout;
* JButton jbtAuthor;
* JButton jbtExit;
* JTextField textField;
* JTextField textFieldResultV;
* JTextField textFieldResultS;

Методы класса:

* MainFrame() - Конструктор класса MainFrame создает элементы рабочего окна, добавляет к каждой кнопке своего слушателя для открытия нужного окна

**Run** - Запускающий класс. Основной при расчете и работе с данными пользователя

Содержит поля

:

* StartPage SP;
* MainFrame MF;
* About Ab;
* Avtor Av;

Методы класса:

* main(String[]) – Выполняется при загрузке программы. Создает экземпляр класса StartPage.
* Arbeit( JTextField numbr, JTextField antwort, JTextArea more) - Вычисление. Прорабатывает методы получения данных пользователя, нахождение данных, заполнение полей формы
* isNumber(String str) – метод проверки строки на число

**StartPage** – стартовая форма, окно «Добро пожаловать». Методы класса:

Методы класса:

* StartPage() - Конструктор класса StartPage. При нажатии на кнопку «Далее», создает экземпляр класса MainFrame.
* LoadImage() – метод загрузки фотографии для фона окна

**BgPanel** – дополнительный класс для отрисовки заднего фона окна

Методы класса:

* paintComponent(Graphics g) – отрисовка фона

**About** – диалоговое окно «О программе». Методы класса:

Содержит поля

* JTextArea jtxtArea
* JButton jbtBack

Методы класса:

* About() - Конструктор класса About создает окно с информацией о прорамме.
* Back() – возврат к предыдущему окну

**Avtor** – диалоговое окно «Об авторе». Методы класса:

Содержит поля:

* JTextArea jtxtArea
* JButton jbtBack
* JLabel foto

Методы класса:

* Avtor() - Конструктор класса Avtor создает окно с информацией об авторе.

# Элементы интерфейса и порядок работы с ним (на примере java-программы)

## Основные функции программы

При загрузке программы появляется стартовое окно (рисунок 9.1). Если фоновая картинка найдена, то при нажатии на кнопку «Далее» открывается главное окно программы (рисунок 9.2).



Рисунок 9.1 – Стартовое окно

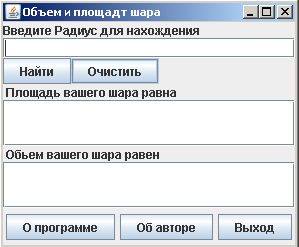


Рисунок 9.2 – Главное окно конвертера единиц объема

Если картинка Background.jpg не найдена, то выдается информационное сообщение (рисунок 9.3). Далее программа работает в обычном режиме – загружается главное окно (рисунок 9.2).

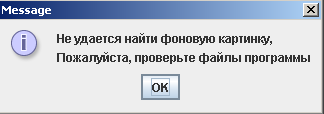


Рисунок 9.3 – Сообщение в случае если файл Background.jpg не найден

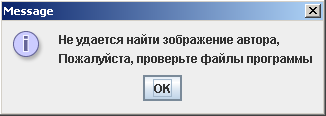


Рисунок 9.3 – Сообщение, если файл Avtor.jpg не найден

Главное окно (рисунок 9.2) содержит поле ввода числа пользователя, поля вывода обьема и площади , кнопки найти, очистить, о программе, о авторе, выйти.

Для работы необходимо ввести положительное число в соответствующее поле и нажать кнопку найти

При нажатии на кнопку «Очистить» все данные из окон ввода, вывода и подробного вывода будут удалены.

В программе предусмотрена обработка ошибок при некорректном вводе данных с выводом соответствующего сообщения (рисунок 9.4).

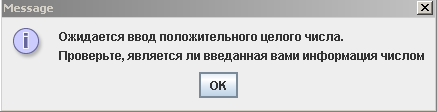


Рисунок 9.4 – Сообщение при вводе некорректных данных

При нажатии кнопок «О программе» и «Об авторе» открываются модальные окна, представленные на рисунках 9.5 и 9.6.

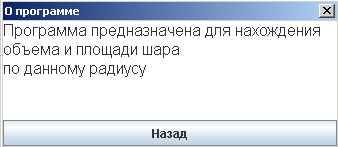


Рисунок 9.5 – Окно «О программе»

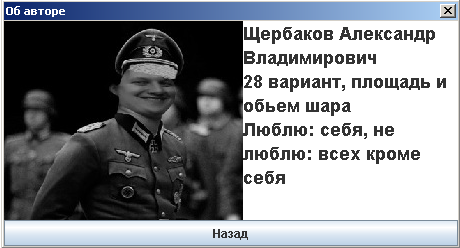


Рисунок 9.6 – Окно «Об авторе»

# UML диаграмма классов

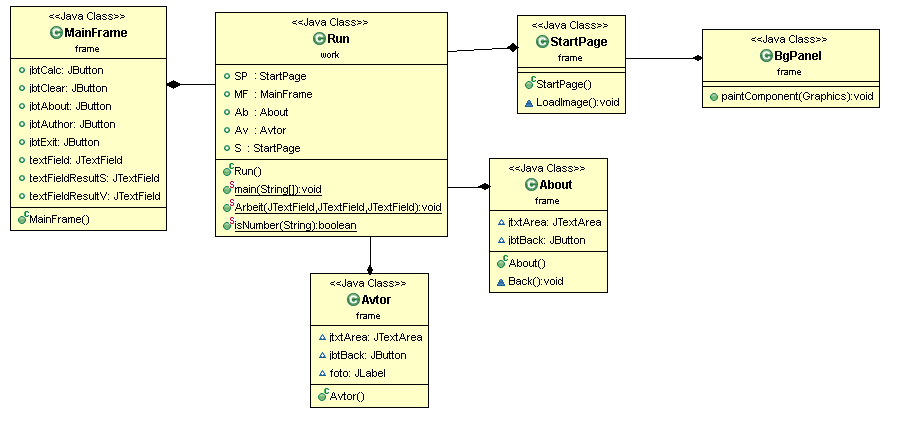


Рисунок 10 – UML диаграмма классов

# Код программы

## Класс Run

**package** work;

**import** javax.swing.JOptionPane;

**import** javax.swing.JTextField;

**import** frame.\*;

/\*\*

\* Главный класс программы. В нем создаются окна и управляют ими, также проводят расчеты

\* **@author** Щербаков

\*

\*/

**public** **class** Run {

//переменные форм

/\*\*

\* Стартовое окно

\*/

**public** **static** StartPage *SP*;

/\*\*

\* Главное окно

\*/

**public** **static** MainFrame *MF*;

/\*\*

\* Программное окно (о программе)

\*/

**public** **static** About *Ab*;

/\*\*

\* Авторское окно

\*/

**public** **static** Avtor *Av*;

/\*\*

\* Главный метод, создает стартовое окно

\* **@param** args строковый массив

\*/

**public** **static** **void** main(String[] args) {

*SP* = **new** StartPage();

*SP*.setVisible(**true**);

}

/\*\*

\* Метод, обьединяющий работу всех методов

\* **@param** numbr текстовое поле, отукда берем радиус

\* **@param** antwort текстовое поле, куда будем записывать ответ обьем

\* **@param** more текстовое поле, в которое будем записывать ответ площадь

\*

\*/

**public** **static** **void** Arbeit( JTextField numbr, JTextField antwort, JTextField more) {

String str = numbr.getText();

**if**(*isNumber*(str) && (Long.*parseLong*(str))!=0) {

**int** R = Integer.*parseInt*(str);

**float** S = (**float**) (4 \* Math.***PI*** \* R \* R);

**float** V = (**float**) (4 \* Math.***PI*** \* R \* R \* R)/3;

more.setText(S+"");

antwort.setText(V+"");

} **else** {

JOptionPane.*showMessageDialog*(**null**, "Ожидается ввод положительного целого числа. \n"+ "Проверьте, является ли введанная вами информация числом");

}

}

/\*\*

\* Метод проверки строки на число

\* **@param** str исходная строка в виде числа

\* **@return** булевое значение, является ли строка числом

\*/

**public** **static** **boolean** isNumber(String str) {

**if** (str == **null** || str.isEmpty()) **return** **false**;

**for** (**int** i = 0; i < str.length(); i++) {

**if** (!Character.*isDigit*(str.charAt(i))) **return** **false**;

}

**return** **true**;

}

}

## Класс StarrPage

**package** frame;

**import** work.\*;

**import** java.awt.BorderLayout;

**import** java.awt.Color;

**import** java.awt.Container;

**import** javax.swing.ImageIcon;

**import** javax.swing.JButton;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** java.awt.Font;

**import** java.awt.Graphics;

**import** java.awt.GridBagLayout;

**import** java.awt.Image;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.io.File;

**import** java.io.IOException;

**import** java.net.URL;

**import** javax.imageio.ImageIO;

**import** javax.swing.JFrame;

**import** javax.swing.JPanel;

**import** javax.swing.JTextArea;

**import** javax.swing.SwingConstants;

/\*\*

\* Стартовое окно программы

\* Создает окно, все панельки, слушателей

\* **@author** - Щербаков

\*

\*/

**public** **class** StartPage **extends** JFrame {

/\*\*

\* **@uml.property** name="frame"

\* **@uml.associationEnd** multiplicity="(1 1)"

\*/

JFrame frame = **new** JFrame("Добро пожаловать");

/\*\*

\* **@uml.property** name="btnNext"

\* **@uml.associationEnd** multiplicity="(1 1)"

\*/

JButton btnNext = **new** JButton("Далее");

/\*\*

\* **@uml.property** name="btnExit"

\* **@uml.associationEnd** multiplicity="(1 1)"

\*/

JButton btnExit = **new** JButton("Выход");

/\*\*

\* **@uml.property** name="pane"

\*/

Container pane = **this**.getContentPane();

**static** Image *im* = **null**;

/\*\*

\* Конструктор создания стартового окна

\* **@throws** IOException

\*/

**public** StartPage() {

setTitle("Добро пожаловать");

setDefaultCloseOperation(***EXIT\_ON\_CLOSE***);

setSize(640, 480);

setLocationRelativeTo(**null**);

//для изображения на фоне устнавливаем перерисовку

setContentPane(**new** BgPanel());

Container cont = getContentPane();

// текст униферситета

JTextArea jtxtUniversity = **new** JTextArea();

jtxtUniversity.setEditable(**false**);

jtxtUniversity.setBackground(**new** Color(255, 255, 255, 0));

jtxtUniversity.setFont(**new** Font("Arial", Font.***PLAIN***, 16));

jtxtUniversity.setForeground(Color.***WHITE***);

jtxtUniversity.setText( " Министерство образования Республики Беларусь\n"

+ "Витебский Государственный Технологический Университет\n"

+ " Кафедра: \"ФИТР\"");

// название курсовой

JTextArea jtxtTheme = **new** JTextArea();

jtxtTheme.setBackground(**new** Color(255, 255, 255, 0));

jtxtTheme.setFont(**new** Font("Arial", Font.***BOLD***, 20));

jtxtTheme.setForeground(Color.***WHITE***);

jtxtTheme.setText("Расчетно-графическая работа по дисциплине \"ООП\"\n"

+ " Объем и площадь шара");

// кто выполнил

JTextArea jtxtWork = **new** JTextArea();

jtxtWork.setBackground(**new** Color(255, 255, 255, 0));

jtxtWork.setFont(**new** Font("Arial", Font.***PLAIN***, 16));

jtxtWork.setForeground(Color.***WHITE***);

jtxtWork.setText("Выполнил :\n"

+ "Щербаков А.В. \n"

+ "Проверил :\n"

+ "Леонов В.В. \n");

// подложка

JLabel lblSityYear = **new** JLabel("Витебск, 2018",SwingConstants.***CENTER***);

lblSityYear.setForeground(Color.***WHITE***);

// разбиваем окно на 4 части

JPanel upFrame = **new** JPanel(**new** GridBagLayout());

JPanel downFrame = **new** JPanel(**new** BorderLayout());

JPanel CenterFrame = **new** JPanel(**new** BorderLayout());

JPanel RigthFrame = **new** JPanel(**new** BorderLayout());

// добавляем каждую часть в общее окно

setLayout(**new** BorderLayout());

pane.setBackground(Color.***WHITE***);

cont.add(upFrame,BorderLayout.***PAGE\_START***);

cont.add(downFrame,BorderLayout.***PAGE\_END***);

cont.add(CenterFrame,BorderLayout.***CENTER***);

cont.add(RigthFrame,BorderLayout.***LINE\_END***);

// про университет вверх, подложку вниз, кнопки по бокам

upFrame.add(jtxtUniversity);

downFrame.add(btnExit, BorderLayout.***LINE\_START***);

downFrame.add(btnNext, BorderLayout.***LINE\_END***);

downFrame.add(lblSityYear, BorderLayout.***PAGE\_END***);

// центрируем название ргр

JPanel u1 = **new** JPanel(**new** GridBagLayout());

u1.add(jtxtTheme);

JPanel c = **new** JPanel(**new** GridBagLayout());

c.add(u1);

CenterFrame.add(c,BorderLayout.***CENTER***);

// добавляем кто сделал вправо

RigthFrame.add(jtxtWork,BorderLayout.***PAGE\_END***);

// кнопка выхода

btnExit.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

Object[] options = { "Выйти", "Остаться" };

**int** n = JOptionPane

.*showOptionDialog*(**null**, "Закрыть окно?",

"Подтверждение", JOptionPane.***YES\_NO\_OPTION***,

JOptionPane.***QUESTION\_MESSAGE***, **null**, options,

options[0]);

**if** (n == 0) {

System.*exit*(0);

}

}

});

// кнопка продолжения

btnNext.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

**if**(Run.*MF* ==**null**) {

Run.*MF* = **new** MainFrame();

}

Run.*MF*.setVisible(**true**);

setVisible(**false**);

}

});

// для большинства панелей включаем прозрачность, для видимости картинки

CenterFrame.setOpaque(**false**);

RigthFrame.setOpaque(**false**);

upFrame.setOpaque(**false**);

downFrame.setOpaque(**false**);

c.setOpaque(**false**);

LoadImage();

}

/\*\*

\* Загрузка заднего фона или вывод сообщения об ошибке

\*/

**void** LoadImage() {

**try** {

URL url = StartPage.**class**.getResource( "/res/Background.jpg");

*im* = **new** ImageIcon(url).getImage();

} **catch** (Exception e) {

JOptionPane.*showMessageDialog*(**this**, "Не удается найти фоновую картинку, \n"

+ "Пожалуйста, проверьте файлы программы");

}

}

}

/\*\*

\* Класс для перерисовки/отображения изображения на стартовой странице

\* **@author** Мр. Хикка

\*

\*/

**class** BgPanel **extends** JPanel{

**public** **void** paintComponent(Graphics g){

g.drawImage(StartPage.*im*, 0, 0, getWidth(), getHeight(), **null**);

}}

## Класс MainFrame

**package** frame;

**import** java.awt.BorderLayout;

**import** java.awt.CardLayout;

**import** java.awt.Container;

**import** java.awt.~~Event~~;

**import** java.awt.FlowLayout;

**import** java.awt.GridBagLayout;

**import** java.awt.GridLayout;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.awt.event.WindowAdapter;

**import** java.awt.event.WindowEvent;

**import** java.awt.event.WindowListener;

**import** java.util.EventObject;

**import** javax.swing.BoxLayout;

**import** javax.swing.DefaultListModel;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JList;

**import** javax.swing.JOptionPane;

**import** javax.swing.JPanel;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTextArea;

**import** javax.swing.JTextField;

**import** javax.swing.border.TitledBorder;

**import** work.Run;

/\*\*

\* Главное окно программы

\* **@version** 1.0 05/05/2018

\* **@author** Щербаков

\*

\*/

**public** **class** MainFrame **extends** JFrame {

/\*\*

\* Метод отрисовки главной формы программы.

\*/

**public** MainFrame() {

// устанавливаем заголовок, выход по крестику, размер и запрещем масщтабировать

setTitle("Объем и площадт шара");

setDefaultCloseOperation(JFrame.***DO\_NOTHING\_ON\_CLOSE***);

setLocationRelativeTo(**null**);

setSize(300,250);

// Создаем кнопки

JButton jbtCalc = **new** JButton("Найти");

JButton jbtClear = **new** JButton("Очистить");

JButton jbtAbout = **new** JButton("О программе");

JButton jbtAuthor = **new** JButton("Об авторе");

JButton jbtExit = **new** JButton("Выход");

// создаем два текстовых поля для ввода и вывода результата

JTextField textField = **new** JTextField(15);

JTextField textFieldResultS = **new** JTextField(15);

JTextField textFieldResultV = **new** JTextField(15);

// Создаем панель кнопок, нижняя часть главного окна

JPanel pBtn = **new** JPanel(**new** FlowLayout());

// Добавляем кнопки на панель кнопок

pBtn.add(jbtAbout);

pBtn.add(jbtAuthor);

pBtn.add(jbtExit);

//панель кнопок найти и очистить

JPanel sc = **new** JPanel();

sc.setLayout(**new** BoxLayout(sc, BoxLayout.***LINE\_AXIS***));

sc.add(jbtCalc);

sc.add(jbtClear);

// создаем панель верхней части окна

JPanel up =**new** JPanel(**new** BorderLayout());

up.add(**new** JLabel("Введите Радиус для нахождения"), BorderLayout.***PAGE\_START***);

up.add(textField,BorderLayout.***CENTER***);

up.add(sc, BorderLayout.***PAGE\_END***);

// создаем центральную часть окна верха

JPanel center = **new** JPanel();

center.setLayout(**new** BoxLayout(center, BoxLayout.***PAGE\_AXIS***));

center.add(**new** JLabel("Площадь вашего шара равна"));

center.add(textFieldResultS);

center.add(**new** JLabel("Обьем вашего шара равен"));

center.add(textFieldResultV);

// создаем общую картину мироздания

add(up,BorderLayout.***PAGE\_START***);

add(center,BorderLayout.***CENTER***);

add(pBtn,BorderLayout.***PAGE\_END***);

// кнопка о программе

jbtAbout.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

**if**(Run.*Ab* ==**null**) {

Run.*Ab* = **new** About();

}

setEnabled(**false**);

Run.*Ab*.setVisible(**true**);

}

});

//кнопка об автаре

jbtAuthor.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

**if**(Run.*Av* ==**null**) {

Run.*Av* = **new** Avtor();

}

setEnabled(**false**);

Run.*Av*.setVisible(**true**);

}

});

// кнопка выйти

jbtExit.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

Object[] options = { "Да", "Нет!" };

**int** n = JOptionPane

.*showOptionDialog*(**null**, "Закрыть окно?",

"Подтверждение", JOptionPane.***YES\_NO\_OPTION***,

JOptionPane.***QUESTION\_MESSAGE***, **null**, options,

options[0]);

**if** (n == 0) {

System.*exit*(0);

}

}

});

// кнопка найти

jbtCalc.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

Run.*Arbeit*(textField, textFieldResultV, textFieldResultS);

}

});

// кнопка очистить

jbtClear.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

textFieldResultS.setText("");

textField.setText("");

textFieldResultV.setText("");

}

});

// предупреждение о выходе

addWindowListener(**new** WindowAdapter() {

@Override

**public** **void** windowClosing(WindowEvent we) {

String ObjButtons[] = {"Да, выйти", "Нет, остаться"};

**int** PromptResult = JOptionPane.*showOptionDialog*(**null**,

"Вы действительно хотите выйти?", "Отвечайте за свои поступки",

JOptionPane.***DEFAULT\_OPTION***, JOptionPane.***WARNING\_MESSAGE***, **null**,

ObjButtons, ObjButtons[1]);

**if** (PromptResult == 0) {

System.*exit*(0);

}

}

});

}

}

## Класс About

**package** frame;

**import** java.awt.BorderLayout;

**import** java.awt.Desktop;

**import** java.awt.Font;

**import** java.awt.GridLayout;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.awt.event.WindowEvent;

**import** java.awt.event.WindowListener;

**import** java.io.File;

**import** java.io.IOException;

**import** javax.swing.JButton;

**import** javax.swing.JDialog;

**import** javax.swing.JOptionPane;

**import** javax.swing.JPanel;

**import** javax.swing.JTextArea;

**import** work.Run;

/\*\*

\* калькулятор

\* Объем и площадь геометрических фигур

\* 28. Объем и площадь шара

\* О программе

\*

\* **@version** 1.0 05/05/2018

\* **@author** Щербаков

\*/

**public** **class** About **extends** JDialog {

/\*\*

\* **@uml.property** name="jtxtArea"

\* **@uml.associationEnd** multiplicity="(1 1)"

\*/

JTextArea jtxtArea = **new** JTextArea();

/\*\*

\* **@uml.property** name="jbtBack"

\* **@uml.associationEnd** multiplicity="(1 1)"

\*/

JButton jbtBack = **new** JButton("Назад");

/\*\*

\* Конструктор класса о программе

\*/

**public** About(){

setSize(340, 150);

setResizable(**false**);

setTitle("О программе");

setLocationRelativeTo(**null**);

setDefaultCloseOperation(***DISPOSE\_ON\_CLOSE***);

// информация о программе

jtxtArea.setColumns(20);

jtxtArea.setRows(5);

jtxtArea.setWrapStyleWord(**true**);

jtxtArea.setLineWrap(**true**);

jtxtArea.setEditable(**false**);

jtxtArea.setText("Программа предназначена для нахождения \n"

+ "объема и площади шара \n"

+ "по данному радиусу");

jtxtArea.setFont(**new** Font("Arial", Font.***PLAIN***, 16));

// панель кнопок

JPanel pnlButton = **new** JPanel(**new** GridLayout(1, 2, 5, 5));

pnlButton.add(jbtBack);

setLayout(**new** BorderLayout());

add(jtxtArea, BorderLayout.***CENTER***);

add(pnlButton, BorderLayout.***SOUTH***);

setFocusable(**true**);

setFocusableWindowState(**true**);

//Кнопка "Назад"

jbtBack.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

Back();

}

});

// При выходе из окна открывать доступ к окну-родителю

addWindowListener(**new** WindowListener() {

**public** **void** windowClosing(WindowEvent event) {

Run.*MF*.setEnabled(**true**);

}

@Override

**public** **void** windowActivated(WindowEvent arg0) {

// **TODO** Auto-generated method stub

}

@Override

**public** **void** windowClosed(WindowEvent arg0) {

// **TODO** Auto-generated method stub

}

@Override

**public** **void** windowDeactivated(WindowEvent arg0) {

// **TODO** Auto-generated method stub

}

@Override

**public** **void** windowDeiconified(WindowEvent arg0) {

// **TODO** Auto-generated method stub

}

@Override

**public** **void** windowIconified(WindowEvent arg0) {

// **TODO** Auto-generated method stub

}

@Override

**public** **void** windowOpened(WindowEvent arg0) {

// **TODO** Auto-generated method stub

}

});

}

/\*\*

\* Возвращение на предыдущее окно

\*/

**void** Back() {

Run.*MF*.setEnabled(**true**);

dispose();

}

}

## Класс Autor

**package** frame;

**import** java.awt.BorderLayout;

**import** java.awt.Font;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.awt.event.WindowEvent;

**import** java.awt.event.WindowListener;

**import** java.io.File;

**import** java.net.URL;

**import** javax.imageio.ImageIO;

**import** javax.swing.ImageIcon;

**import** javax.swing.JButton;

**import** javax.swing.JDialog;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JTextArea;

**import** work.Run;

/\*\*

\*калькулятор

\* Объем и площадь геометрических фигур

\* 28. Объем и площадь шара

\* О программе

\*

\* **@version** 1.0 05/05/2018

\* **@author** Щербаков

\*/

**public** **class** Avtor **extends** JDialog{

/\*\*

\* **@uml.property** name="jtxtArea"

\* **@uml.associationEnd** multiplicity="(1 1)"

\*/

JTextArea jtxtArea = **new** JTextArea();

/\*\*

\* **@uml.property** name="jbtBack"

\* **@uml.associationEnd** multiplicity="(1 1)"

\*/

JButton jbtBack = **new** JButton("Назад");

/\*\*

\* **@uml.property** name="foto"

\* **@uml.associationEnd** multiplicity="(1 1)"

\*/

JLabel foto;

/\*\*

\* Конструктор окна об авторе

\*/

**public** Avtor() {

setTitle("Об авторе");

setSize(460, 250);

setDefaultCloseOperation(***DISPOSE\_ON\_CLOSE***);

setResizable(**false**);

setLocationRelativeTo(**null**);

/\*\*

\* Загрузка изображения, ели его нет - вывод предупредительного окна

\*/

**try** {

URL url = StartPage.**class**.getResource( "/res/Avtor.jpg");

foto = **new** JLabel(**new** ImageIcon(url)) ;

} **catch**(Exception e) {

JOptionPane.*showMessageDialog*(**this**, "Не удается найти изображение автора, \n"

+ "Пожалуйста, проверьте файлы программы");

}

// текст про автора

jtxtArea.setEditable(**false**);

jtxtArea.setText("Щербаков Александр Владимирович \n"

+ "28 вариант, площадь и обьем шара \n"

+ "Люблю: себя, не люблю: всех кроме себя");

jtxtArea.setColumns(20);

jtxtArea.setRows(5);

jtxtArea.setWrapStyleWord(**true**);

jtxtArea.setLineWrap(**true**);

jtxtArea.setFont(**new** Font("Dialog", Font.***BOLD***, 18));

// установка компанентов на фрейм

setLayout(**new** BorderLayout());

add(foto, BorderLayout.***WEST***);

add(jtxtArea, BorderLayout.***CENTER***);

add(jbtBack, BorderLayout.***SOUTH***);

// Кнопка назад

jbtBack.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

Run.*MF*.setEnabled(**true**);

dispose();

}

});

// фокусируемся на текущем фрейме

setFocusable(**true**);

setFocusableWindowState(**true**);

// при закрытии окна, разрешать пользоваться другим

// и ради этого переопределяем методы

addWindowListener(**new** WindowListener() {

**public** **void** windowClosing(WindowEvent event) {

Run.*MF*.setEnabled(**true**);

}

@Override

**public** **void** windowActivated(WindowEvent arg0) {

// **TODO** Auto-generated method stub

}

@Override

**public** **void** windowClosed(WindowEvent arg0) {

// **TODO** Auto-generated method stub

}

@Override

**public** **void** windowDeactivated(WindowEvent arg0) {

// **TODO** Auto-generated method stub

}

@Override

**public** **void** windowDeiconified(WindowEvent arg0) {

// **TODO** Auto-generated method stub

}

@Override

**public** **void** windowIconified(WindowEvent arg0) {

// **TODO** Auto-generated method stub

}

@Override

**public** **void** windowOpened(WindowEvent arg0) {

// **TODO** Auto-generated method stub

}

});

}

}

# Список использованных источников

1. Хорстманн, К.С. Java 2. Библиотека профессионала, том1. Основы. 8-е издание, : Пер. с англ. / К. С. Хорстманн, Г. Корнелл - М.: ООО «Вильямс», 2012. - 816 с.
2. Дирк, Л. Самоучитель Java 7: Пер. с нем. / Л. Дирк, П. Мюллер. – СПб: БХВ-Петербург, 2013. – 464 с.: илл.
3. Шилдт, Г. Java. Полное руководство. 8-е издание, : Пер. с англ. / Г. Шилдт. – М.:ООО «Вильямс», 2012. – 1104 с.
4. Блинов, И.Н. Java. Промышленное программирование./ И.Н. Блинов, В.С. Романчик –Минск: «Четыре четверти», 2013. – 896 с.