Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации»

Нижегородский институт управления

Кафедра Информатики и информационных технологий

ОТЧЕТ

Задания лекции №15

Выполнила: студент группы:

ИБ-321

Кодоева Нели Тамазиевна

Нижний Новгород

2023 г.

Оглавление

[**Задание 1** 3](#_Toc135740555)

[**Ответы на вопросы:** 3](#_Toc135740556)

[**Задание 2** 4](#_Toc135740557)

[**Ответы на вопросы:** 4](#_Toc135740558)

[**Задание 3** 5](#_Toc135740559)

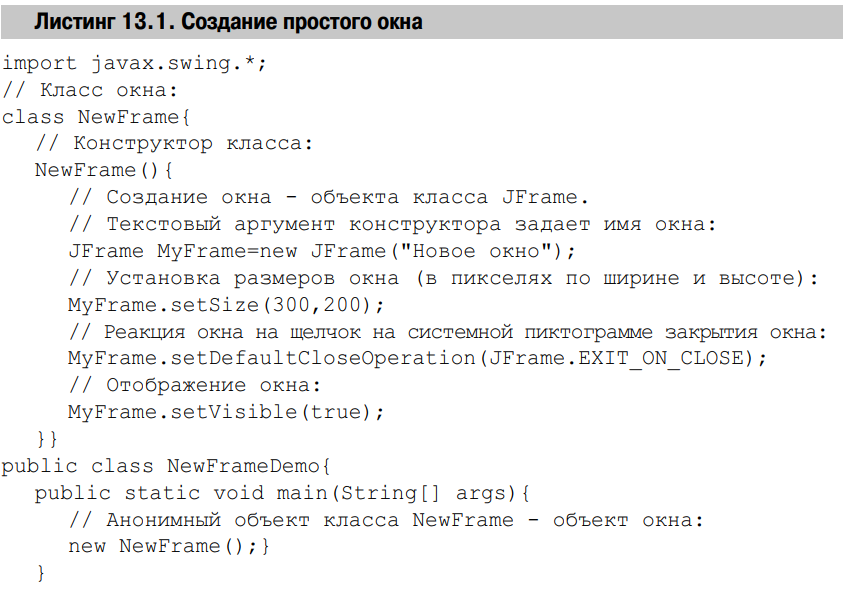
[**Классы событий** 5](#_Toc135740560)

[**Интерфейсы** 6](#_Toc135740561)

# **Задание 1**

**Вопросы к листингу 13.1**

1. **Сколько классов создаются в листинге?**
2. **Какой из них основной (базовый)?**
3. **Что вызывает построение окна на экране монитора?**
4. **Какая строчка создаёт объект класса NewFrame?**
5. **Что такое системные пиктограммы визуального объекта?**
6. **Что значит «Закрыть визуальный объект на экране монитора»?**
7. **Как решается эта задача в данном листинге (какой метод и аргументы используются)?**
8. **Какие основные этапы отображения визуального объекта на экране монитора можно выделить?**



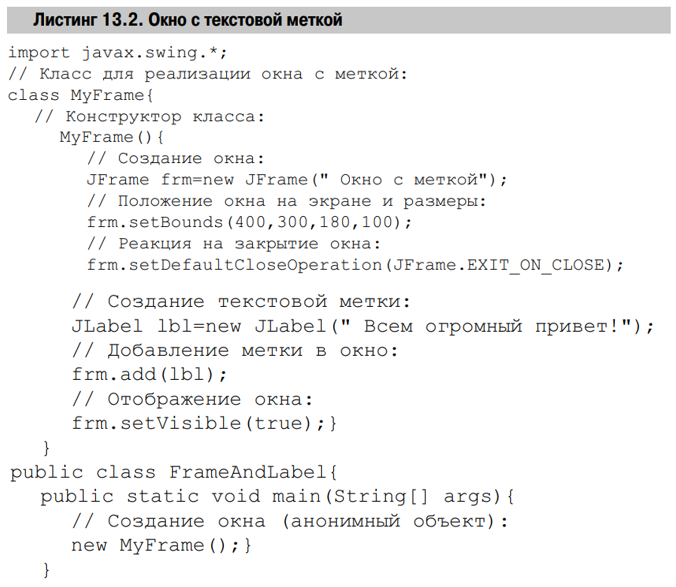
**Ответы на вопросы:**

1. В листинге создаётся два класса NewFrame и NewFrameDemo;
2. NewFrameDemo – базовый класс;
3. Построение окна на экране монитора вызывает строка MyFrame.setVisible(true);
4. Объект класса NewFrame создаёт строка newFrame() в классе NewFrameDemo;
5. Системными пиктограммами визуального объекта являются системные иконки;
6. «Закрыть визуальный объект на экране монитора» - сокрытие визуального объекта или прекращение действия;
7. Данная задача решается при помощи строки MyFrame.setDefaultCloseOperation (JFrame.EXIT\_ON\_CLOSE);
8. Создаем экземпляр класса NewFrame. Далее создаем в конструкторе экземпляра класса JFrame и выделяем место под окно. После отображаем окно при помощи метода setVisible().

# **Задание 2**

**Вопросы к листингу 13.2**

1. **Почему (где) создаётся и к какому классу относится объект frm?**
2. **Чем (каков статус) является строка “ Окно с меткой”?**
3. **Перечислить методы класса Jframe, использованные в данном листинге (**[**https://java-online.ru/swing-windows.xhtml**](https://java-online.ru/swing-windows.xhtml) **).**
4. **Особенности метода добавления компонентов?**
5. **Что такое анонимный объект класса? (**[**https://javarush.com/groups/posts/2193-anonimnihe-klassih**](https://javarush.com/groups/posts/2193-anonimnihe-klassih) **)?**
6. **Какой код приводит к появлению изображения на экране монитора?**



**Ответы на вопросы:**

1. Объект frm создается в строке JFrame frm=new JFrame(" Окно с меткой") и относится к классу JFrame;
2. «Окно с меткой» - выводимое название окна;
3. Методы класса Jframe: setBounds(), setDefaultCloseOperation(), add(), setVisible();
4. Метод вызывается из того объекта, в который компонент добавляется, а аргументом методу передается добавляемый компонент;
5. Анонимный объект класса - это объект, который создается без присвоения ему имени;
6. К появлению изображения на экране монитора приводит строка frm.setVisible(true)

# **Задание 3**

**Создать для себя шпаргалку по классам событий с элементами графического интерфейса в Java**

## **Классы событий**

| **Класс событий** | **Интерфейс прослушивателя** | **Описание** |
| --- | --- | --- |
| ActionEvent | ActionListener | Событие, которое указывает, что произошло определенное компонентом действие, такое как нажатие кнопки или выбор элемента из списка элементов меню. |
| AdjustmentEvent | AdjustmentListener Настройка списка | Событие настройки генерируется настраиваемым объектом, таким как полоса прокрутки. |
| ComponentEvent | ComponentListener | Событие, которое указывает, что компонент переместился, изменился размер или изменилась его видимость. |
| ContainerEvent | ContainerListener | Когда компонент добавляется в контейнер (или) удаляется из него, это событие генерируется объектом контейнера. |
| FocusEvent | FocusListener | Это события, связанные с фокусировкой, которые включают focus, focusin, focusout и blur. |
| ItemEvent | ItemListener | Событие, которое указывает, был ли выбран элемент или нет. |
| Ключевое событие | KeyListener | Событие, которое возникает из-за последовательности нажатий клавиш на клавиатуре. |
| MouseEvent | MouseListener и MouseMotionListener | События, которые происходят из-за взаимодействия пользователя с мышью (указывающее устройство). |
| MouseWheelEvent | Указатель поворота мыши | Событие, которое указывает, что колесо мыши было повернуто в компоненте. |
| Текстовое событие | TextListener | Событие, которое происходит при изменении текста объекта. |
| WindowEvent | WindowListener | Событие, которое указывает, изменило ли окно свой статус или нет. |

## **Интерфейсы**

| **Интерфейс прослушивателя** | **Методы** |
| --- | --- |
| **ActionListener** | * **actionPerformed()** |
| AdjustmentListener Настройка списка | * adjustmentValueChanged() |
| ComponentListener | * componentResized() * componentShown() * componentMoved() * componentHidden() |
| ContainerListener | * componentAdded() * componentRemoved() |
| FocusListener | * focusGained() * focusLost() |
| ItemListener | * itemStateChanged() |
| KeyListener | * Тип ключа() * Нажатие клавиши() * Выпущенный ключ() |
| MouseListener | * Нажата кнопка мыши() * Щелчок мышью() * mouseEntered() * mouseExited() * mouseReleased() |
| MouseMotionListener для мыши | * Перемещение мышью() * Перехваченный мышью() |
| Указатель поворота мыши | * mouseWheelMoved() |
| TextListener | * TextChanged() |
| WindowListener | * Активировано окно() * windowDeactivated() Активирован * Открыто окно() * windowClosed() * Закрытие окна() * windowIconified() * windowDeiconified() |