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

ПО ЗАДАНИЮ К ЛЕКЦИИ №15

«Приложения с графическим интерфейсом»

Выполнил: студент группы: ИБ-321

Борисков Дмитрий Андреевич

Преподаватель: Окулич Виктор Иванович

Нижний Новгород

2023 г.

Оглавление

[**Задание 15.1 Ответить на вопросы** 3](#_Toc133966370)

[Вопросы к листингу 13.1 3](#_Toc133966371)

[Рисунок 1. Листинг 13.1 3](#_Toc133966372)

[Ответы на вопросы к листингу 13.1 3](#_Toc133966373)

[**Задание 15.2 Ответить на вопросы** 4](#_Toc133966374)

[Вопросы к листингу 13.2 4](#_Toc133966375)

[Рисунок 2. Листинг 13.2 4](#_Toc133966376)

[Ответы на вопросы к листингу 13.2 4](#_Toc133966377)

[**Задание 15.3 Создать для себя шпаргалку по классам событий с элементами графического интерфейса в Java** 5](#_Toc133966378)

[Классы событий в Java 5](#_Toc133966379)

[Интерфейсы и их методы 6](#_Toc133966380)

[Поток обработки событий 7](#_Toc133966381)

[Подходы к кодированию 8](#_Toc133966382)

# **Задание 15.1 Ответить на вопросы**

## 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?

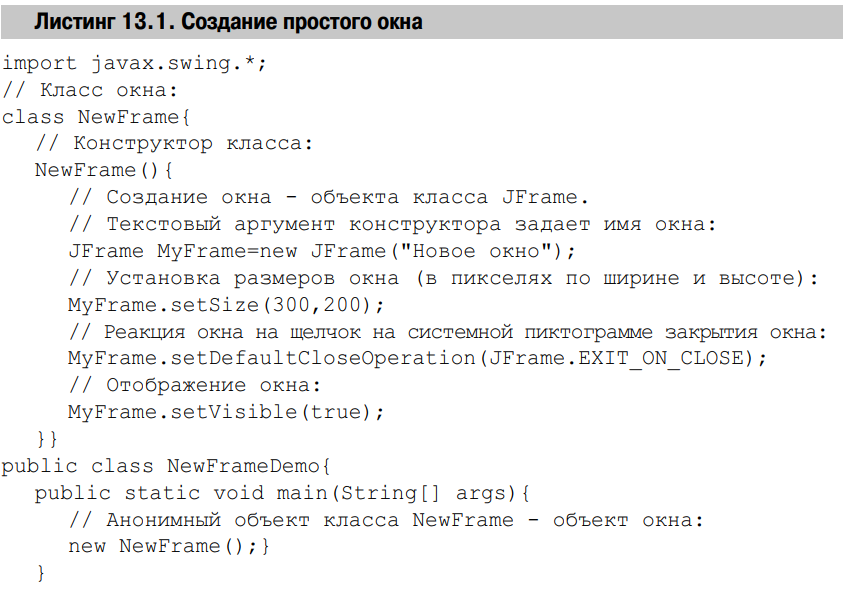


Рисунок 1. Листинг 13.1

## Ответы на вопросы к листингу 13.1

1. В листинге создаётся два класса NewFrame и NewFrameDemo;
2. Базовым является класс NewFrameDemo;
3. Построение окна вызывает строка MyFrame.setVisible(true);
4. Объект класса NewFrame создаёт строка newFrame() в классе NewFrameDemo;
5. Системные пиктограммы визуального объекта, это системные иконки;
6. Закрыть визуальный объект на экране монитора, означает его сокрытие или прекращение действия;
7. Данная задача решается при помощи строки MyFrame.setDefaultCloseOperation (JFrame.EXIT\_ON\_CLOSE);
8. Сначала создаётся экземпляр класса NewFrame. Далее идёт создание в конструкторе экземпляра класса JFrame и выделение места под окно. После всего этого, идёт отображение окна при помощи метода setVisible().

# **Задание 15.2 Ответить на вопросы**

## 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> ).
4. Особенности метода добавления компонентов?
5. Что такое анонимный объект класса? (<https://javarush.com/groups/posts/2193-anonimnihe-klassih> )?
6. Какой код приводит к появлению изображения на экране монитора?

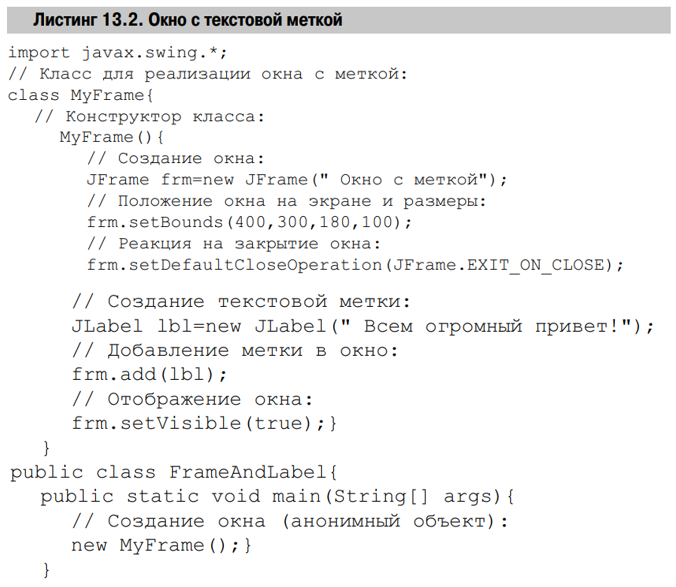


Рисунок 2. Листинг 13.2

## Ответы на вопросы к листингу 13.2

1. Объект frm создается в строке JFrame frm=new JFrame(" Окно с меткой") и относится к классу JFrame;
2. Выводимое название окна;
3. Методы класса Jframe:
   1. setBounds();
   2. setDefaultCloseOperation();
   3. add();
   4. 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 15.3 Создать для себя шпаргалку по классам событий с элементами графического интерфейса в Java**

## Классы событий в Java

| **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. |

Таблица . Классы событий

## Интерфейсы и их методы

| **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() |

Таблица . Интерфейсы и их методы

## Поток обработки событий

* Для генерации события требуется взаимодействие пользователя с компонентом;
* Объект соответствующего класса событий создается автоматически после генерации события и содержит всю информацию об источнике события;
* Вновь созданный объект передается методам зарегистрированного прослушивателя;
* Метод выполняется и возвращает результат.

## Подходы к кодированию

Три подхода к выполнению обработки событий заключаются в размещении кода обработки событий в одном из указанных ниже мест.

1. Внутри класса;
2. Другой класс;
3. Анонимный класс.