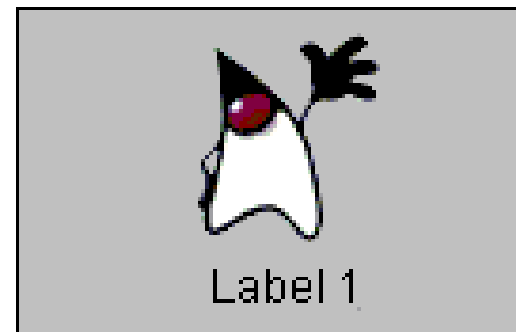
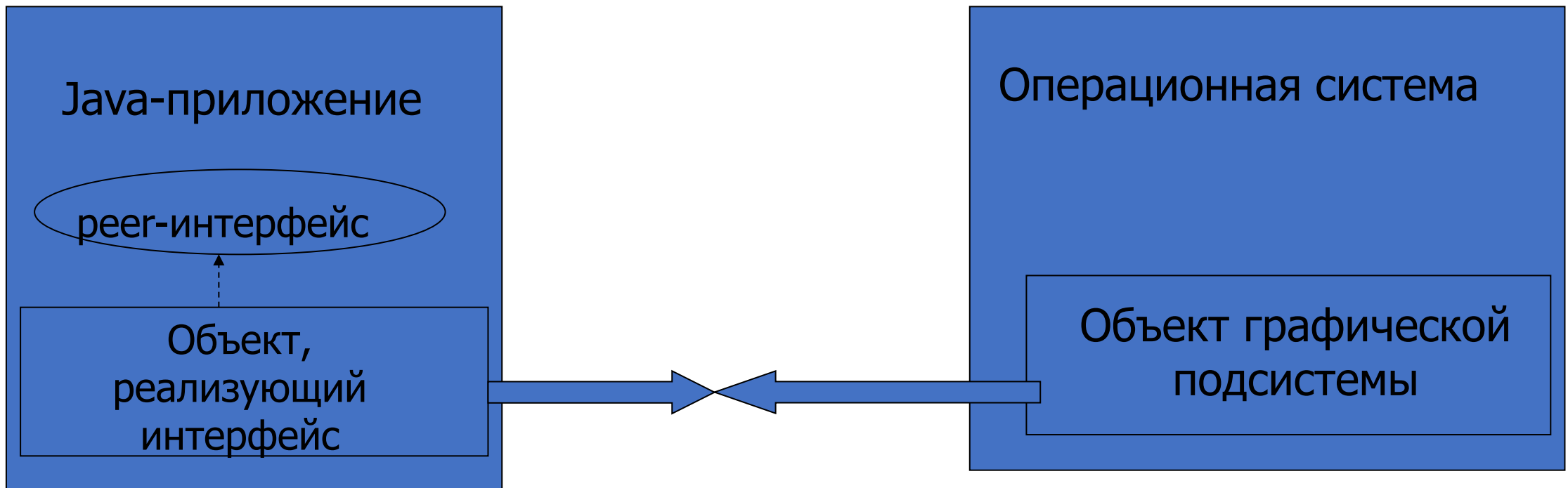


# Средства создания графического интерфейса в Java



# Библиотека AWT

- peer-to-peer интерфейсы («тяжелые» компоненты)



# «Легкие» (lightweight) компоненты

- Сохранение заданного при создании вида (look and feel)
- Возможность изменить вид в любой момент работы приложения (PL&F)
- Библиотека «легких» компонентов Swing

# JavaFX

- 2007г. – альтернатива Flash (Sun Microsystems)
- JavaFX 8 вошла в JDK(Oracle)
- Использует язык разметки FXML
- Начиная с Java 11 версии, The Java FX не входит в JavaSE (компания Gluon)
- Ресурсы
  - <https://openjfx.io/>
  - <https://gluonhq.com/products/javafx/>
  - <https://habr.com/ru/articles/474292/>

# Приложение JavaFX

```
public class Main extends Application{  
    public static void main(String[] args) {  
        launch(args);  
    }  
    @Override  
    public void start(Stage stage) {  
        Label text = new Label("Hello !");  
        Pane root = new Pane(text);  
        Scene scene = new Scene(root,300,250);  
        stage.setScene(scene);  
        stage.setTitle("JavaFX Application");  
        stage.show();  
    }  
}
```

# javafx.application.Application

- `init()`: инициализирует приложение до его запуска. Метод не должен использоваться для создания графического интерфейса или отдельных его частей.
- `start(Stage stage)`: здесь определяется графический интерфейс.
- `stop()`: вызывается после закрытия приложения

# ПОТОКИ

- Ни метод `init`, ни конструктор класса, который наследуется от `Application`, не подходит и не должен использоваться для создания графического интерфейса. Потому что и метод `init`, и конструктор запускаются в потоке, который называется потоком запуска или `launcher thread`.
- А создание и изменение графического интерфейса должно производиться в потоке приложения или `application thread`. Именно в таком потоке и запускается метод `start` (и метод `stop`).



# ПОТОКИ ВЫПОЛНЕНИЯ

```
public static void main(String[] args) {  
    System.out.println("Метод main()" + " " +  
        Thread.currentThread().getName());  
    Application.launch(args);  
}
```

@Override

```
public void init() throws Exception {  
    System.out.println("Метод init()" + " " +  
        Thread.currentThread().getName());  
}
```

@Override

```
public void stop() throws Exception {  
    System.out.println("Метод stop()" + " " +  
        Thread.currentThread().getName());  
}
```

@Override

```
public void start(Stage stage) throws Exception {  
    System.out.println("Метод start()" + " " +  
        Thread.currentThread().getName());
```

```
    BorderPane root = new BorderPane();  
    Label text = new Label("Привет, мир!");  
    text.setFont(new Font(32));  
    root.setCenter(text);  
    Scene scene = new Scene(root, 400, 150);
```

```
    stage.setTitle("Заголовок окна");  
    stage.setScene(scene);  
    stage.show();
```

```
}
```

# Структура приложения

- Каждое JavaFX приложение состоит из иерархии нескольких основных компонентов:
  - Stage (окна или подмости)
  - Scene (сцены)
  - Node (узлы)

# Stage

Stage -> Window->Object

Stage()

Stage(StageStyle s)

В JavaFX application создается первичное окно, которое передается в метод start()

# Окно по нажатию кнопки

```
Button bt = new Button("Создать окно");
root.setCenter(bt);
bt.setOnAction(event-> {
    newWind();
});
// добавляем метод создания окна newWind()
public void newWind() {
    Stage wind = new Stage();
    wind.setTitle("Новое окно");
    wind.show();
}
```

# Стили

StageStyle.DECORATED

StageStyle.UNDECORATED

StageStyle.TRANSPARENT

StageStyle.UTILITY

```
Stage w = new Stage(StageStyle.DECORATED);
```

```
Stage w = new Stage();  
w.initStyle(StageStyle.UTILITY);
```

# Управление

<https://openjfx.io/javadoc/18/javafx.graphics/javafx/stage/Stage.html>

По умолчанию открывается документация для 21



# Модальность

Modality.NONE

Modality.WINDOW\_MODAL

Modality.APPLICATION\_MODAL

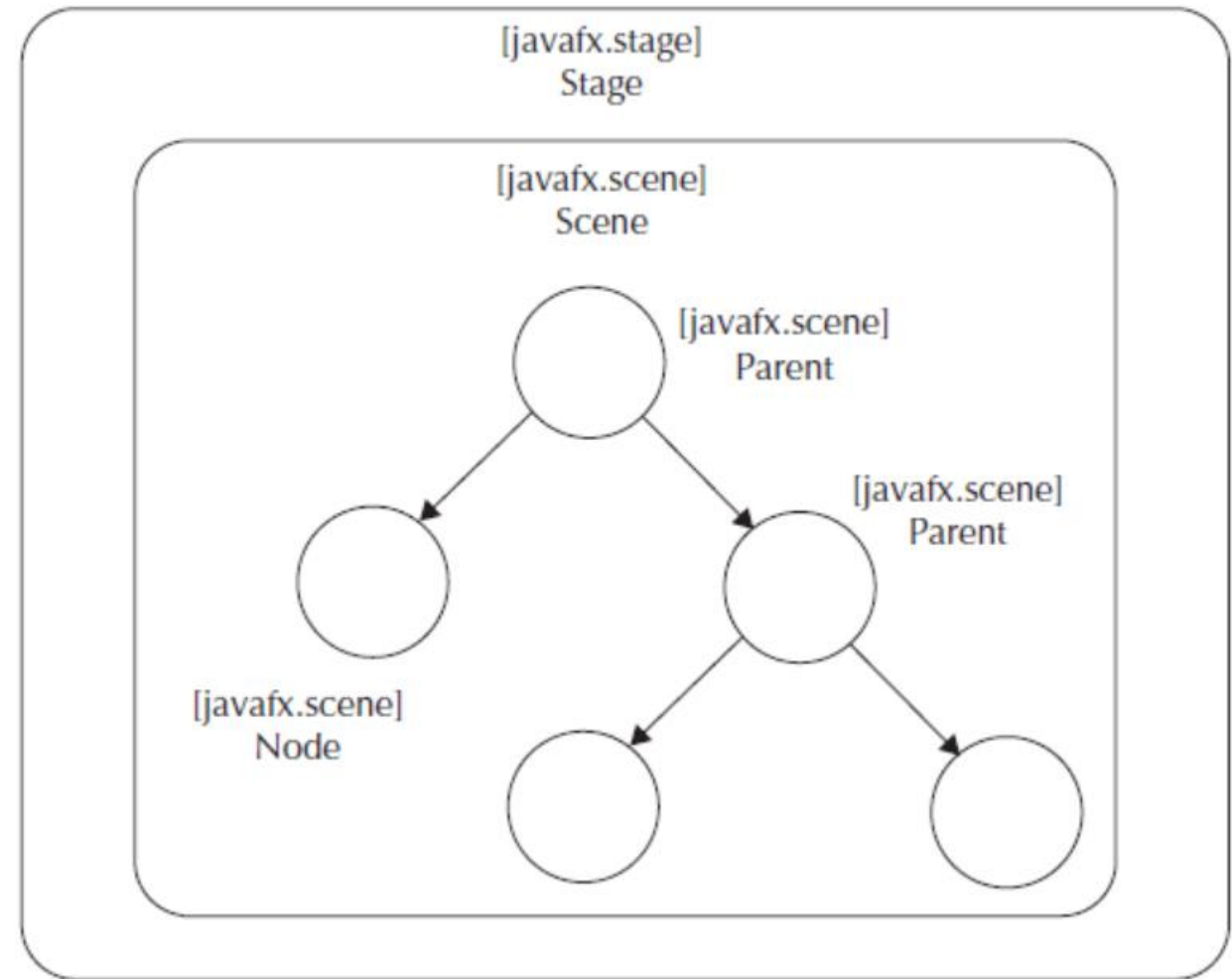
```
Stage s = new Stage();
```

```
s.initModality(Modality.WINDOW_MODAL);
```

```
s.initOwner(parentStage);
```

# Scene

- `javafx.scene.Parent`
- `javafx.scene.Node`



- <https://openjfx.io/javadoc/18/javafx.graphics/javafx/scene/package-summary.html>

# Scene

Scene(Parent root)

Scene(Parent root, double width, double height)

Scene(Parent root, Paint fill)

Scene(Parent root, double width, double height, Paint fill)

Несколько примеров

@Override

```
public void start(Stage stage) {  
    Group root = new Group();  
    Scene scene = new Scene(root, 200, 150);  
    scene.setFill(Color.LIGHTGRAY);  
    Circle circle = new Circle(60, 40, 30, Color.GREEN);  
    Text text = new Text(10, 90, "JavaFX Scene");  
    text.setFill(Color.DARKRED);  
    Font font = new Font(20);  
    text.setFont(font);  
    root.getChildren().add(circle);  
    root.getChildren().add(text);  
    stage.setScene(scene);  
    stage.show();  
}
```

@Override

```
public void start(Stage stage) {  
    Label label = new Label("Hello");           // текстовая метка  
    Button button = new Button("Button");        // кнопка  
    Group group = new Group(button);             // вложенный узел Group  
    FlowPane root = new FlowPane(label, group);  // корневой узел  
    Scene scene = new Scene(root, 300, 150);      // создание Scene  
    stage.setScene(scene);                      // установка Scene для Stage  
    stage.setTitle("Hello JavaFX");  
    stage.show();  
}
```

@Override

```
public void start(Stage stage) {  
    Group root = new Group();  
    Scene scene = new Scene(root, 400, 150, Color.BLUE);  
    //scene.setFill(Color.BLUE); альтернативная установка цвета  
    stage.setScene(scene);  
    stage.setTitle("Hello JavaFX");  
    stage.show();  
}
```