

Буфер обмена

Работа с текстом

Spinner<T>

TextInputControl базовый абстрактный класс

TextFormatter

TextField

PasswordField

TextArea

HTMLEditor

WebView

WebEngine

Методы

```
public final void setText (String t)
```

```
public final String getText ()
```

```
public final String getSelectedText (String t)
```

```
public void copy()
```

```
public void cut()
```

```
public void paste()
```

Примеры

- Файлы
HTMLMain.java

Clipboard

```
// Модуль javafx.graphics
```

Статический метод

```
Clipboard cb = Clipboard.getSystemClipboard();
```

ClipboardContent

Данные, добавляемые в буфер обмена

```
ClipboardContent content = new ClipboardContent();
```

Наследует HashMap <DataFormat, Object>

Добавление данных

- `putString(String s)`– добавляет данные `DataFormat.PLAIN_TEXT`

```
Clipboard cb = Clipboard.getSystemClipboard();  
ClipboardContent content = new ClipboardContent();  
content.putString("Текст, добавляемый в буфер обмена");  
cb.setContent(content);  
//очистить буфер cb.clear();
```

Добавление данных

- `putHtml(String h)` добавляет данные `DataFormat.HTML`
- `putRTF(String rtf)` добавляет данные `DataFormat.RTF`
- `putUrl(String url)` добавляет данные `DataFormat.URL`
- `putFiles(List<File> f)` добавляет данные `DataFormat.FILES`
- `putFilesByPath(List<String> f)` добавляет данные `DataFormat.FILES`
- `putImage(Image img)` добавляет данные `DataFormat.IMAGE`
- `put ()` добавляет данные пользовательского типа

Все методы возвращают `boolean`

Проверка типа данных в буфере

`boolean hasString()`

`boolean hasHtml()`

`boolean hasRtf()`

`boolean hasUrl()`

`boolean hasFiles()`

`boolean hasImage()`

`boolean hasContent()`

Получение данных из буфера

String getString()

String getHtml()

String getRTF()

String getUrl()

List<File> String getFiles()

Image getImage()

Object getObject(Object dataFormat)

Пример

- Файлы

Clipbrd.java

image.png

Drag & drop

DragEvent событие перетаскивания и сброса

Три этапа:

- Запуск
- Перемещение и сброс
- Проверка результата

Drag & drop

- **TransferMode** - определяет тип операции (COPY, MOVE, LINK, ANY)
- **Dragboard** - контейнер для передаваемых данных
- **ClipboardContent** - хранилище данных (может содержать текст, изображения, файлы)
- **setOnDragDetected** - инициализация перетаскивания
- **setOnDragOver** - проверка возможности сброса
- **setOnDragDropped** - обработка сброса данных

TransferMode

- Перечисление (enum), которое определяет, какие действия разрешены при перетаскивании данных между источником и целью.
- Это ключевой элемент для контроля поведения Drag&Drop

```
public enum TransferMode {  
    COPY,    // Копирование данных  
    MOVE,    // Перемещение данных  
    LINK,    // Создание ссылки/связи  
    ANY      // Любое действие (COPY, MOVE или LINK)  
}
```

TransferMode.MOVE

- Данные перемещаются из источника в цель.
- Источник должен удалить свои данные после успешного перемещения.

Запуск перетаскивания

```
labelSrc.addEventHandler(MouseEvent.DRAG_DETECTED, event -> {  
    System.out.println("DRAG_DETECTED");  
    Dragboard db =  
        labelSrc.startDragAndDrop(TransferMode.COPY);  
    ClipboardContent content = new ClipboardContent();  
    content.putString(labelSrc.getText());  
    db.setContent(content);  
    event.consume();  
});
```

Проверка результата

```
labelSrc.addEventHandler(DragEvent.DRAG_DONE, event -> {  
    System.out.println("DRAG_DONE");  
    if (event.isAccepted()) {  
        System.out.println("Сброс выполнен успешно.");  
    }  
    else {  
        System.out.println("Сброс не выполнен");  
    }  
    System.out.println("Режим: " + event.getTransferMode());  
});
```

Обработка сброса

```
labelRecip.addEventHandler(DragEvent.DRAG_OVER, event -> {  
    Dragboard db = event.getDragboard();  
    if (db.hasString() || db.hasHtml() || db.hasRtf()) {  
        event.acceptTransferModes(TransferMode.COPY);  
    }  
    event.consume();  
});
```

Обработка сброса

```
labelRecip.addEventHandler(DragEvent.DRAG_DROPPED, event -> {  
    Dragboard db = event.getDragboard();  
    boolean success = false;  
    if (db.hasString()){  
        labelRecip.setText(db.getString())  
        success = true;  
    }  
    event.setDropCompleted(success);  
    event.consume();  
});
```

Информация о событии

`acceptTransferModes()` указать режимы, поддерживаемые при сбросе

`getTransferMode()` узнать режим сброса

`isAccepted()` результат сброса

`getDragboard()` ссылка на буфер

`getX()`, `getY()`, `getZ()` координаты сброса относительно узла

`getSceneX()`, `getSceneY()` координаты относительно экрана

Dragboard

- Буфер, используемый для хранения данных при перетаскивании
- Наследует Clipboard и все методы этого класса
- Позволяет дополнительно задать изображение под указателем мыши при перетаскивании

`setDragView(Image im)` значок перетаскивания

Пример

- Файлы

SimpleDragDrop.java

FileDragDrop.java

DnD.java

blue32x32.png