ОБРАБОТКА СОБЫТИЙ

Назначить обработчик событий

- Методом с названием setOnXXX, изменяющим свойство onXXX

```
Button bt = new Button (“Press”);
bt.setOnAction( event -> {
        System.out.println(“Нажата кнопка Press”);
});
```

Отменить обработчик

```
bt.setOnAction(null);
```

Назначить обработчик событий

- Методами addEventFilter() и addEventHandler()
- Можно удалять removeEventFilter() и removeEventHandler()

- Порядок срабатывания для Button

  Filter

  Handler

  setOnAction

Назначить обработчик событий

- addEventFilter() можно зарегистрировать несколько

```
bt.addEventFilter(ActionEvent.ACTION, event->{
        System.out.println(“Нажата кнопка Press”);
        if ( ???) {
                System.out.println(“Больше не нажимать”);
                event.consume();
        }
});
```

Назначить обработчик событий

- addEventHandler() можно зарегистрировать несколько

```
bt.addEventHandler(ActionEvent.ACTION, event->{
        System.out.println(“Нажата кнопка Press”);
});
```

Назначить обработчик событий

- Если нужно отследить изменения свойства

```
stage.xPropert().addListener((obj, oldV, newV) ->{
        System.out.println(“Движемся по X ”+newV);
});
```

В метод регистрации можно передать

- Анонимный вложенный класс, реализующий интерфейс EventHandler

```
bt.setOnAction( new EventHandler<ActionEvent>()
@Override
 public void handle(ActionEvent event) {
        System.out.println(“Нажата кнопка Press”);
});
```

- Лямбда-выражение

```
bt.setOnAction( event -> {
        System.out.println(“Нажата кнопка Press”);
});
```

В метод регистрации можно передать

- Объект класса, реализующего интерфейс EventHandler

```
private EventHandler<ActionEvent> handl1 = event -> {
        System.out.println(“Нажата кнопка Press”);
};
bt.setOnAction(handl1);
```

- Ссылку на метод, получающий объект- событие XXX

```
private
 void onClick(ActionEvent event){
        System.out.println(“Нажата кнопка Press”);
};
bt.setOnAction(this::onClick);
```

Движение события

- Методы addEventFilter() и addEventHandler() имеют все узлы, объекты сцены и окна
- Если назначить обработчик нажатия кнопки для контейнера, то он перехватит нажатие всех кнопок в контейнере
- Событие движется сначала от окна до источника, вызвавшего события, а затем обратно к окну

Пример

- Кнопка, контейнер, сцена и окно имеют обработчики событий
  - stage – Filter
  - scene – Filter
  - root – Filter
  - button – Filter
  - button – Handler
  - button – OnAction
  - root – Handler
  - scene – Handler
  - stage – Handler

Генерация события

- Если класс реализует интерфейс EventTarget, ему можно послать событие программным путем

```
bt.setOnAction( event -> {
        System.out.println(“Нажата кнопка Press”);
        System.out.println(“генерируем событие нажатия другой кнопки”);
        Event.fireEvent(otherBt, event);
});
```

Классы событий

ActionEvent

InputEvent

- KeyEvent
- MouseEvent
  - MouseDragEvent
- TouchEvent
- GestureEvent
- DragEvent
- ContexMenuEvent

WindowEvent

События окна

- WindowEvent
  - WindowEvent. ANY
  - WindowEvent. WINDOW_SHOWING
  - WindowEvent. WINDOW_SHOWN
  - WindowEvent. WINDOW_CLOSE_REQUEST
  - WindowEvent. WINDOW_HIDING
  - WindowEvent. WINDOW_HIDDEN

События окна

- Изменение состояния окна
  - stage.showingProperty()
  - stage.focusedProperty()
  - stage.xProperty()
  - stage.yProperty()
  - stage.widthProperty()
  - stage.heightProperty()
  - stage.iconofiedProperty()
  - stage.maximisedProperty()
  - stage.fullScreenProperty()

События клавиатуры

- KeyEvent
  - KeyEvent.ANY
  - KeyEvent.KEY_PRESSED
  - KeyEvent.KEY_RELEASED
  - KeyEvent.KEY_TYPED

Получение информации

event.getCode()

event.getText()

event.getCharacter()

event.isShiftDown()

event.isControlDown()

event.isAltDown()

События мыши

- MouseEvent
  - MouseEvent.ANY
  - MouseEvent.MOUSE_PRESSED
  - MouseEvent.MOUSE_RELEASED
  - MouseEvent.MOUSE_CLICKED
  - MouseEvent.MOUSE_MOVED
  - MouseEvent.MOUSE_DRAGGED
  - MouseEvent.MOUSE_ENTERED
  - MouseEvent.MOUSE_EXITED

Прокрутка

- ScrollEvent
  - ScrollEvent.ANY
  - ScrollEvent.SCROLL
  - ScrollEvent.SCROLL_STARTED
  - ScrollEvent.SCROLL_FINISHED