Objektovo orientované programovanie

Učiteľ:
Ing. Jozef Wagner, PhD.

Učebnica:
https://oop.wagjo.com/

OPG
Teória 25

1. Property objekt
2. Prepojenie - binding
3. Transformácia property objektov

Opakovanie

Na sledovanie zmeny hodnoty používame interface `ObservableValue`

Počúvanie na zmenu hodnoty sa robí pomocou metódy `addListener()`

Takýto objekt sa volá **`property`**, získame ho väčšinou pomocou metódy `valueProperty()`

```
slider.valueProperty().addListener(
    (observable, oldValue, newValue) -> {
    ...
  });
```

javafx.beans.property.Property

Property obalí (wrap) bežnú hodnotu (int, String, atď.) a pridá ku nej ďalšie schopnosti:

- **Výpočet aktuálnej hodnoty** - `getValue`
- **Zmena hodnoty** - `setValue`
- **Change listeners** - upozornenie pri zmene hodnoty - `addListener`
- **Binding** - prepojenie viacerých properties - `bind` a `bindBidirectional`

Property objekty v komponentoch

Komponenty poskytujú property pre svoje vnútorné hodnoty

- `textProperty()` - text (`String`) ak má komponent nejaký text (`Label`, `TextField`, `TextArea`, ...)
- `valueProperty()` - číslo (`Double`) pri komponentoch s číselnou hodnotou (`Slider`)
- `rotateProperty()` - stupeň natočenia (`Double`), má ho takmer každý komponent
- `disableProperty()` - `boolean`, či je komponent neaktívny (`Button`, `CheckBox`)
- `selectedProperty()` - `boolean`, či je komponent vybraný (`CheckBox`, `RadioButton`)

Vlastné property objekty

Základné abstraktné typy property objektov sú:

- `DoubleProperty` - uchováva reálnu číselnú hodnotu
- `IntegerProperty` - uchováva celočíselnú hodnotu
- `StringProperty` - uchováva textovú hodnotu
- `BooleanProperty` - uchováva booleovskú hodnotu

Nové objekty vytvoríme pomocou '`Simple`' tried (`SimpleDoubleProperty`, `SimpleIntegerProperty` atď.)

```
IntegerProperty vekProperty = new SimpleIntegerProperty(18);
```

```
BooleanProperty isStudentProperty = new SimpleBooleanProperty(true);
```

Prepojenie property objektov - Binding

Binding - Najdôležitejšia funkcionalita property objektov

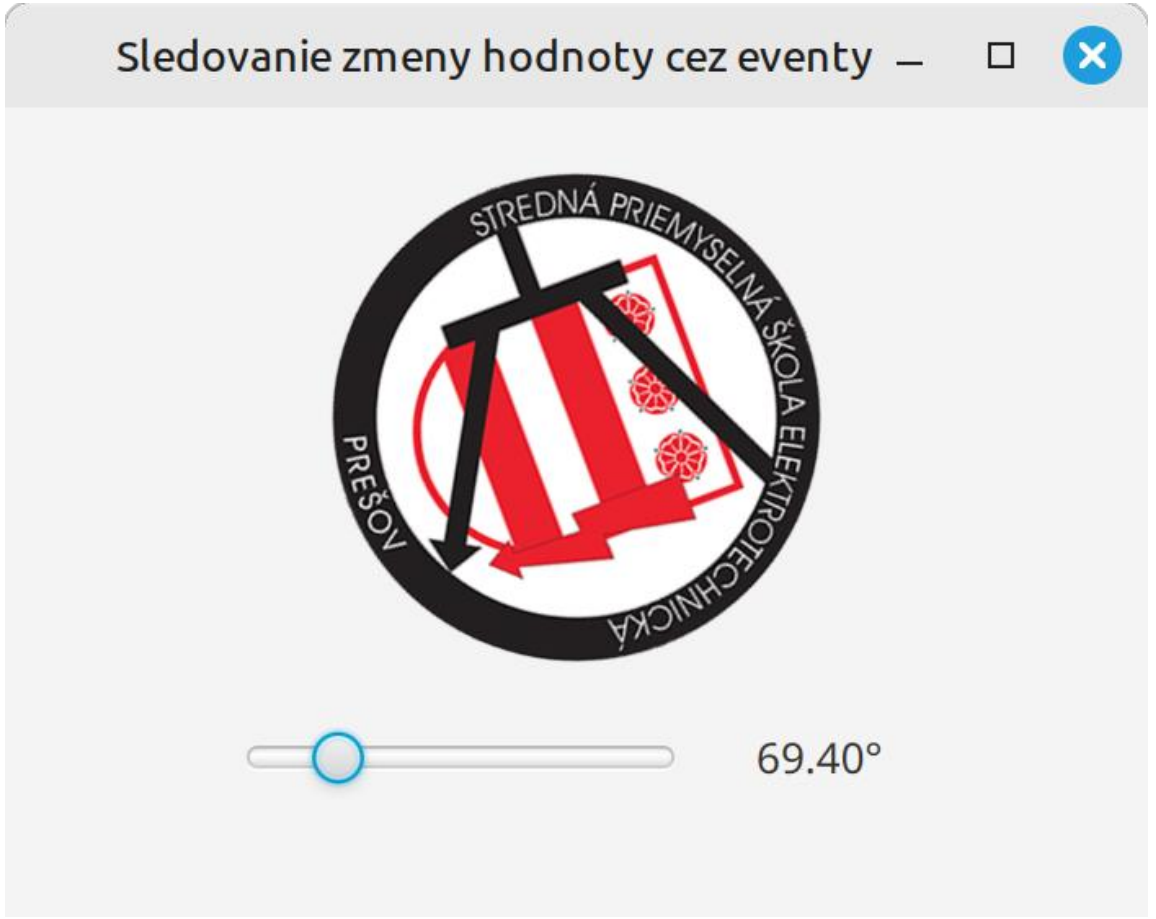
Zmena hodnoty v property sa automaticky prejaví na prepojených property

- **jednosmerné prepojenie** - unidirectional bind - zmena propagovaná jedným smerom
- **obojsmerné prepojenie** - bidirectional bind - zmena propagovaná oboma smermi

Prepojenie sa robí pomocou `bind()` resp. `bindBidirectional()`

Príklad: `fooProperty.bind(barProperty);`

Keď sa zmení hodnota `barProperty` tak sa automaticky zmení aj hodnota `fooProperty`. *Druhým smerom to nebude platiť*.



V tomto príklade sú pre nás zaujímavé nasledovné property objekty:

- Slider má property so svojou hodnotou - `slider.ValueProperty()` - Double
- Komponent s logom má property nad hodnotou otočenia - `obrazok.rotateProperty()` - Double
- Label má property s textom - `label.textProperty()` - String

Namiesto:

```
public void initialize() {
  slider.valueProperty().addListener((obs, oldVal, newVal) -> {
    obrazok.setRotate(newVal.doubleValue());
  });
}
```

Použijeme prepojenie:

```
public void initialize() {
    obrazok.rotateProperty().bind(slider.valueProperty());
}
```

Ostáva nám prepojiť label, ale Double property sa nedá priamo prepojiť so String property

Transformácia property objektov

javafx.beans.binding.`Bindings` poskytuje metódy na transformáciu hodnôt

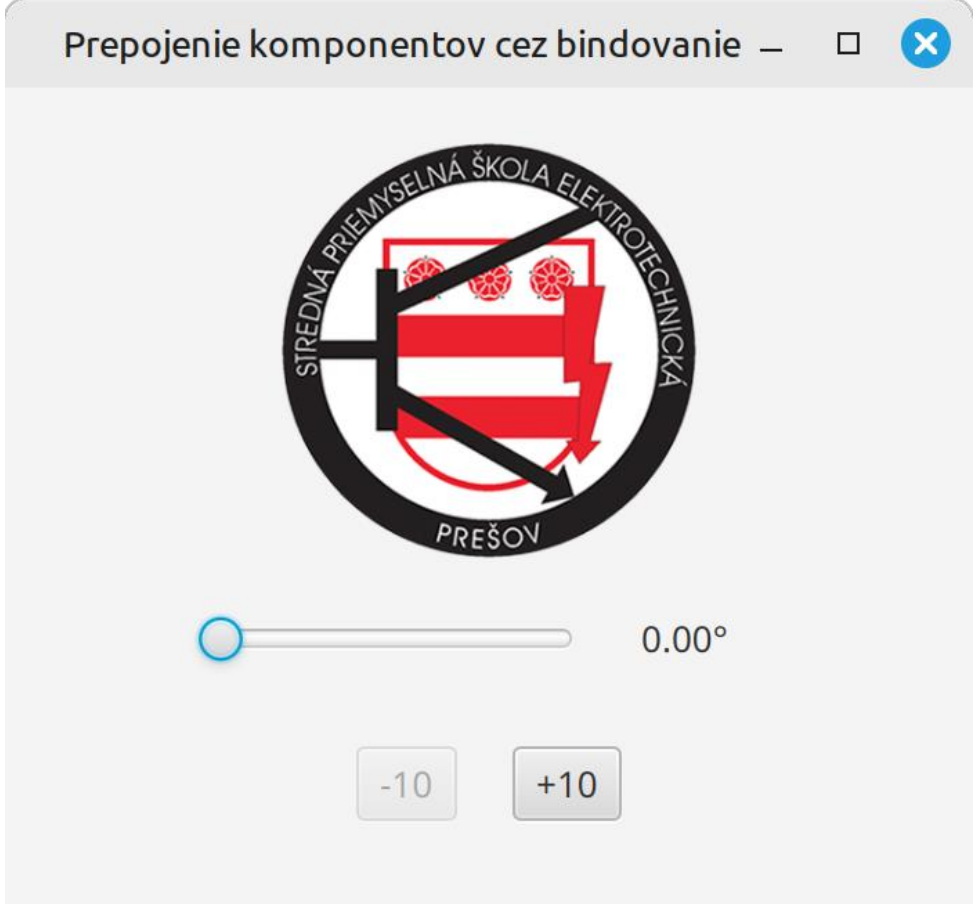
- Manipulácie s číslami: `add`, `subract`, `multiply`, `divide`, `negate`, `max`, `min`.
- Prevod na text: `format` - funguje podobne ako String.format
- Prevod na booleovskú hodnotu: `notEqual`, `lessThan`, `greaterThan`.

Príklady:

```
Bindings.add(10, property)
```

```
Bindings.notEqual(property1, property2)
```

Transformácie vieme voľne kombinovať



- Slider má property so svojou hodnotou - `slider.ValueProperty()` - Double
- Komponent s logom má property nad hodnotou otočenia - `obrazok.rotateProperty()` - Double
- Label má property s textom - `label.textProperty()` – String
- *Tlačidlá majú property neaktívnosti* - `plusButton.disableProperty()` - *Boolean*

```
public void initialize() {

    obrazok.rotateProperty().bind(slider.valueProperty());

    label.textProperty().bind(
      Bindings.format("%.2f°", slider.valueProperty()));

    minusButton.disableProperty().bind(
      Bindings.lessThanOrEqual(slider.valueProperty(),0));

    plusButton.disableProperty().bind(
      Bindings.greaterThanOrEqual(slider.valueProperty(), 360));

}
```


Prepojenie komponentov cez bindovanie
STREDNÁ PRIEMYSELNÁ ŠKOLA ELEKTROTECHNICKÁ
PREŠOV
30.00°
-10
+10

```
public void initialize() {
    obrazok1.rotateProperty().bind(
        Bindings.divide(slider.valueProperty(), 4.0));
    obrazok2.rotateProperty().bind(
        Bindings.divide(slider.valueProperty(), 2.0));
    obrazok3.rotateProperty().bind(
        slider.valueProperty());
    obrazok4.rotateProperty().bind(
        Bindings.multiply(slider.valueProperty(), 2.0));

    label.textProperty().bind(
        Bindings.format("%.2f°", slider.valueProperty()));
    minusButton.disableProperty().bind(
        slider.valueProperty().lessThanOrEqualTo(0));
    plusButton.disableProperty().bind(
        slider.valueProperty().greaterThanOrEqualTo(360));
}
```