

## Exercise 2

In java, beans often represent real world data. Represented as object with which you can interact, define different properties and they can be observed. They are good for handling and observing changes, as well as acting on these changes and properties. They are often used to modify and store data, for later use.

A Property in JavaFX is a convention, after which some objects may follow. It defines the getter, setter and a new function in JavaFX, the `...Property()`. As an example, here is a short implementation of a property:

```
import javafx.beans.property.DoubleProperty;
import javafx.beans.property.SimpleDoubleProperty;
class Temperature {

    // Define a variable to store the property
    private DoubleProperty temper = new SimpleDoubleProperty();

    // Define a getter for the property's value
    public final double getTemper(){return temper.get();}

    // Define a setter for the property's value
    public final void setTemper(double value){temper.set(value);}

    // Define a property getter for the property itself
    public DoubleProperty temperProperty() {return temper;}
}
```

the `temperProperty` returns the whole object, not just a value of it. Events that change on property changes can be implemented with the `ChangeListener`:

```
public class Main{
    public static void main(String[] args) {
        Temperatur temper = new Temperatur(10);

        temper.temperProperty()
            .addListener(new ChangeListener(){e ->
                System.out.print("Temperatur changed");})
    }
}
```

Bindings are used to bind to variables or objects together, so they always depend on each other

(or just in one way). In the following example from <https://docs.oracle.com/javafx/2/binding/jfxpub-binding.htm> the sum is depending on the two values of num1 and num2:

```
package bindingdemo;

import javafx.beans.property.IntegerProperty;
import javafx.beans.property.SimpleIntegerProperty;
import javafx.beans.binding.NumberBinding;

public class Main {

    public static void main(String[] args) {
        IntegerProperty num1 = new SimpleIntegerProperty(1);
        IntegerProperty num2 = new SimpleIntegerProperty(2);
        NumberBinding sum = num1.add(num2);
        System.out.println(sum.getValue());
        num1.set(2);
        System.out.println(sum.getValue());
    }
}
```

To enable a button only if the length of the input in a given field is greater or equal than x, you simply bind the length value of the input to the `isUsable` (or similar) property of the Button. Further you could bind the property `isSelected` of a Radiobutton group to the same button, so the button is only useable if there are at least x symbols in the text field and one of the options of the radiobuttons is selected.

source: <https://docs.oracle.com/javafx/2/binding/jfxpub-binding.htm>