

Komunikacja między komponentami

Zadanie 1.

- Stwórz projekt o nazwie **LessonTwo**.
- Utwórz komponent **InnerOne**.
- Utwórz komponent **InnerTwo**.
- W komponencie **App** niech będzie pole tekstowe a wartość, którą wpisze tam użytkownik będziemy aktualizować co każdy znak w zmiennej **text** typu *String*.
- W komponencie **InnerOne** zaimportuj dekorator **@Input** z **@angular/core**, edytując w poniższy sposób pierwszą linię pliku **inner-one.component.ts**:

```
import { Component, OnInit, Input } from '@angular/core';
```
- W komponencie **InnerOne** utwórz zmienną **receivedText** typu *String* z dekoratorem **@Input()**:

```
@Input() receivedText: String;
```
- W pliku **inner-one.component.html** dodaj linię:

```
<h1>{{ receivedText }}</h1>
```
- W pliku **app.component.html** dodaj linię:

```
<app-inner-one [receivedText]="text"></app-inner-one>
```
- Przetestuj działanie aplikacji w przeglądarce.

Zadanie 2.

- W komponencie **InnerOne** zaimportuj dekorator **@Output** z **@angular/core** oraz klasę **EventEmitter** edytując w poniższy sposób pierwszą linię pliku **inner-one.component.ts**:

```
import { Component, OnInit, Input, Output, EventEmitter } from '@angular/core';
```
- W pliku **inner-one.component.ts** dodaj zmienną **out**:

```
@Output() out: EventEmitter<String> = new EventEmitter();
```

- W komponencie **InnerOne** dodaj przycisk, na kliknięcie którego zostanie wywołana metoda `sendText()`. Ciało tej metody znajduje się poniżej:

```
sendText(): void {
    this.out.emit(this.receivedText + ' was sent. ');
}
```
- W pliku **app.component.ts** dodaj zmienną `received` typu `String`.
- W pliku **app.component.ts** dodaj metodę `onOut()`:

```
onOut(received: String): void {
    this.received = received;
}
```
- W kodzie HTML komponentu **App** zmodyfikuj linię związaną z `<app-inner-one>`:

```
<app-inner-one [receivedText]="text"
(out)="onOut($event)"></app-inner-one>
```
- W kodzie HTML komponentu **App** dodaj linię:

```
<h3>{{ received }}</h3>
```
- Przetestuj działanie aplikacji w przeglądarce.

Zadanie 3. (ocenie)

Zaimplementuj wzajemną komunikację między komponentami **InnerOne** oraz **InnerTwo** w następujący sposób:

- W komponencie **InnerOne** będzie pole tekstowe, a wartość tego pola zapisywana w jakiejś zmiennej oraz przycisk.
- W komponencie **InnerTwo** jest element **H4**, który wyświetla zawartość pola tekstowego z komponentu **InnerOne**, po wcześniejszym kliknięciu na przycisk.
- Analogicznie w komponencie **InnerTwo** będzie pole tekstowe i przycisk a w komponencie **InnerOne** element **H4**, który wyświetla wartość tego pola.
- Do samej komunikacji należy użyć komponentu **App** (w nim oczywiście umieścić komponent **InnerTwo**) jak w poprzednich zadaniach.

Rozwiązanie zadania należy umieścić na repozytorium GitHuba w katalogu EXERCISE03 a ten katalog w katalogu ANGULAR-CW02.

Zadanie 4. (oceniane)

Zapoznaj się z materiałem dotyczącym **WebStorage API HTML5** a następnie zaimplementuj komunikację między komponentami **InnerOne** oraz **InnerTwo** jak opisano w **zadaniu 3.** z tą różnicą, że teraz **zamiast** używać dekoratorów *Input* i *Output* użyjesz **WebStorage API HTML5**.

Rozwiązanie zadania należy umieścić na repozytorium GitHuba w katalogu EXERCISE04 a ten katalog w katalogu ANGULAR-CW02.