W jaki sposób komponenty się komunikują?

- **Dekorator @Input()** – komponent rodzic przekazuje dane do komponentu dziecka
- **Dekorator @Output()** – komponent dziecko emituje dane do komponentu rodzica za pomocą EventEmitter() i wywołania .emit()
- **Serwis (singleton)**
- Subskrypcja do współdzielonego obiektu **Subject/BehaviorSubject** i wywołanie .next()
- **Dekorator @ViewChild()**
- Subskrypcja do **EventEmitter()** – niezalecane! EventEmitter() powinien służyć do emitowania wartości z komponentów, a nie do ręcznego subskrybowania się do niego.

@Input()

- Za pomocą tego dekoratora komponent rodzic (parent) może przekazać dane do komponentu dziecka (child). Dekorator przyjmuje opcjonalną nazwę do użycia np. @Input('nazwa-wlasciwosci'). Konieczny jest import **Input** z **@angular/core** jak na załączonym poniżej obrazku.

child-component.component.ts

```ts
import {Component, Input} from '@angular/core';

5+ usages
@Component({
  selector: 'app-child-component',
  templateUrl: './child-component.component.html',
  styleUrls: ['./child-component.component.css']
})
export class ChildComponentComponent {
  @Input() title!: string;
  @Input('my-id') id!: string;
}
```

child-component.component.html

```html
<p>Wyświetlam moje {{ id }}</p>
<p>Tytuł: {{ title }}</p>
```

app.component.html

```html
<app-child-component [my-id]="'10'" [title]="'Testowy Input'"></app-child-component>
<app-child-component my-id="10" title="Testowy Input"></app-child-component>
```

Wyświetlam moje 10

Tytuł: Testowy Input

Wyświetlam moje 10

Tytuł: Testowy Input

@Output()

- Za pomocą tego dekoratora komponent dziecko (child) może komunikować się z komponentem rodzicem (parent). Dekorator przyjmuje opcjonalną nazwę do użycia np. @Output('my-event'). Konieczny jest import **Output** oraz **EventEmitter** z **@angular/core**.

child-component.component.ts

```ts
import {Component, EventEmitter, Output} from '@angular/core';

5+ usages
@Component({
  selector: 'app-child-component',
  templateUrl: './child-component.component.html',
  styleUrls: ['./child-component.component.css']
})
export class ChildComponentComponent {
  @Output() callParent = new EventEmitter<string>();
  1 usage
  heyParentComponent(): void {
    this.callParent.emit( value: 'Cześć rodzic!')
  }
}
```

child-component.component.html

```html
<button (click)="heyParentComponent()">Wywołaj rodzica</button>
```

app.component.html

```html
<app-child-component (callParent)="reactToChildCall($event)"></app-child-component>
```

```ts
export class AppComponent {
  1 usage
  title = 'pierwszy-projekt';

  1 usage
  reactToChildCall(message: string) {
    console.log(message)
  }
}
```