

Uvod u Angular

NIKOLA AJZENHAMER

PROGRAMIRANJE ZA VEB

MATEMATIČKI FAKULTET

2020/2021

Sadržaj

Uvod

Komponente

Uvod

Osnovne informacije

Razvojno okruženje za kreiranje naprednih klijentskih veb aplikacija

- Struktuiranje i stilizovanje
- Centralizacija odgovornosti usluga
- Rad sa formularima
- Navigacija na klijentu
- Slanje asinhronih HTTP zahteva
- Animacije
- ...

Verzisionisanje

- Angular.js – verzija Angular 1 (<https://angularjs.org/>)
- Angular – verzija Angular 2+ (<https://angular.io/>)

Angular CLI

Instalacija: `npm install -g @angular/cli`

Testiranje instalacije: `ng --version`

Kreiranje novog projekta: `ng new naziv-projekta`

- Rutiranje?
- Mašina za stilizovanje?

Pokretanje servera za razvoj: `ng serve`

- <http://localhost:4200/>

Uključivanje Semantic UI biblioteke

1. Otvoriti `index.html` datoteku

2. U sadržaj elementa `head` dodati naredne linije

```
<link rel="stylesheet"
      href="https://cdn.jsdelivr.net/npm/semantic-ui@2.4.2/dist/semantic.min.css" />
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/semantic-ui@2.4.2/dist/semantic.min.js"></script>
```

Uključivanje Bootstrap 4 biblioteke

1. Instalacija: `npm install bootstrap jquery popper.js`
2. Otvoriti `angular.json` datoteku
3. U niz `projects.naziv-projekta.architect.build.options.styles` dodati nisku `"node_modules/bootstrap/dist/css/bootstrap.min.css"`
4. U niz `projects.naziv-projekta.architect.build.options.scripts` dodati niske `"node_modules/jquery/dist/jquery.min.js"`,
`"node_modules/popper.js/dist/popper.min.js"`,
`"node_modules/bootstrap/dist/js/bootstrap.min.js"`

Komponente

Osnovni koncepti

Komponenta – osnovna gradivna jedinica svih Angular aplikacija

Komponente koje čine jednu Angular aplikaciju organizuju hijerarhiju

Najčešće je u korenu te hijerarhije komponenta [AppComponent](#)

- Automatski se kreira prilikom kreiranja novog Angular projekta

Kreiranje hijerarhije komponenti

Primer aplikacije

Hypatia Demo

Odjavi se

Kursevi

Programiranje za veb

Baze podataka

Uvod u veb i internet tehnologije

Razvoj softvera

Programiranje za veb

Smer: Informatika

Godina studija: 4

ESPB: 6

Odaberite ispitni rok:

Jun 1

Pristupi sadržaju

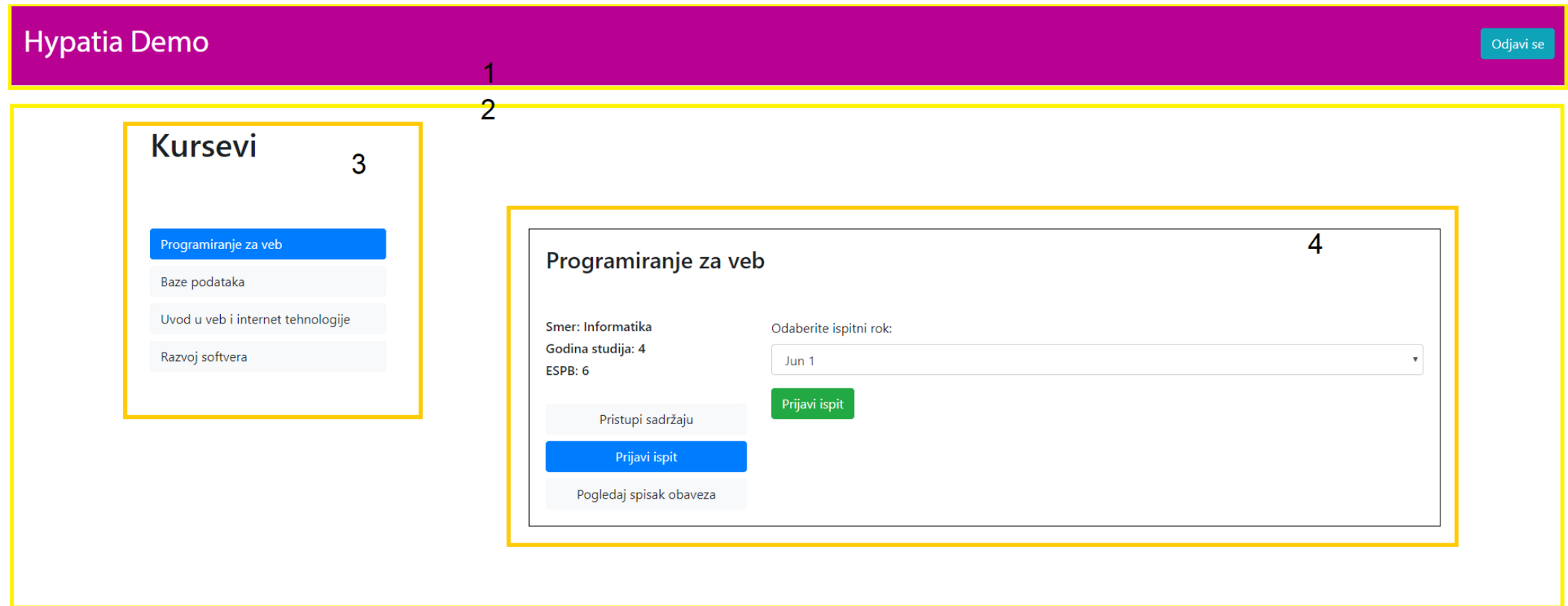
Prijavi ispit

Pogledaj spisak obaveza

Prijavi ispit

Kreiranje hijerarhije komponenti

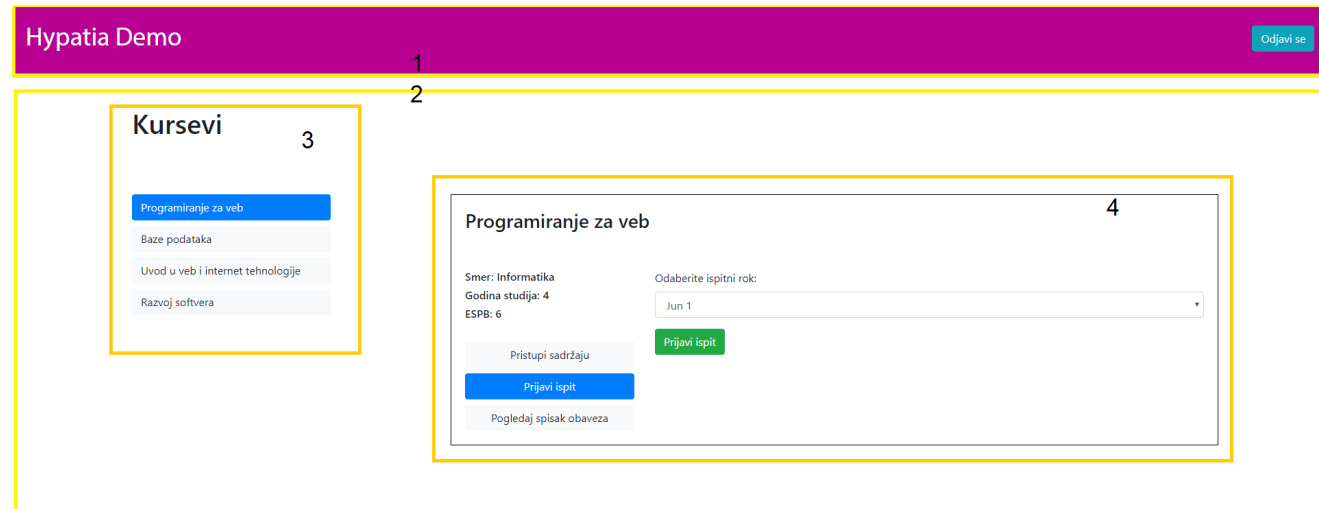
Podela aplikacije na komponente – primer 1



Kreiranje hijerarhije komponenti

Podela aplikacije na komponente
Primer 1

1. Zaglavlje stranice
2. Glavni deo
3. Odabir kursa
4. Upravljanje informacijama o kursu

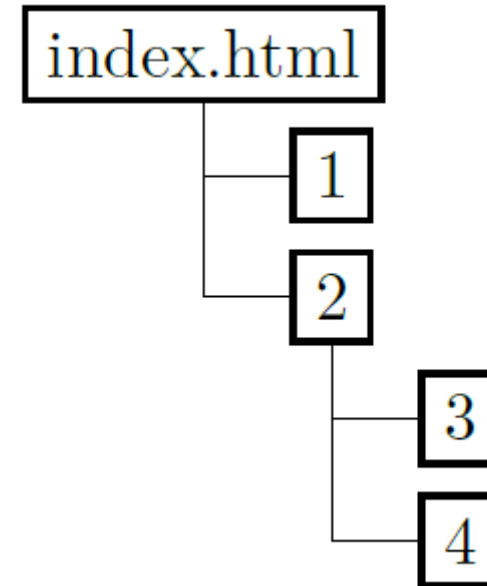


Kreiranje hijerarhije komponenti

Podela aplikacije na komponente

Primer 1

1. Zaglavlje stranice
2. Glavni deo
3. Odabir kursa
4. Upravljanje informacijama o kursu



Kreiranje hijerarhije komponenti

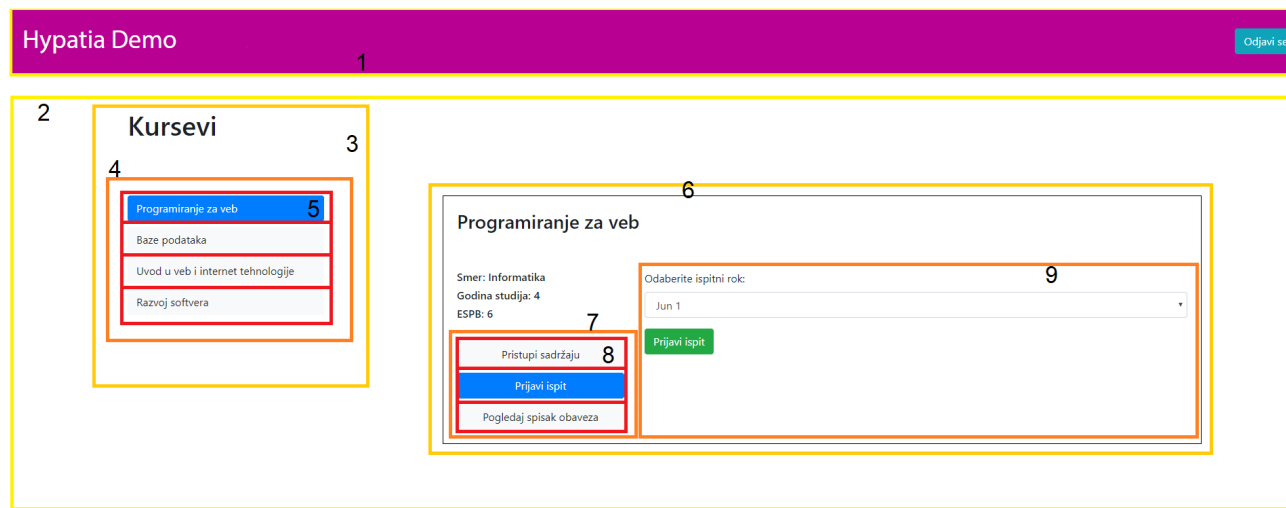
Podela aplikacije na komponente – primer 2



Kreiranje hijerarhije komponenti

Podela aplikacije na komponente
Primer 2

1. Zaglavlje stranice
2. Glavni deo
3. Upravljanje kursevima
4. Lista kurseva
5. Odabir kursa
6. Upravljanje informacijama o kursu
7. Lista opcija
8. Odabir opcije
9. Detalji opcije

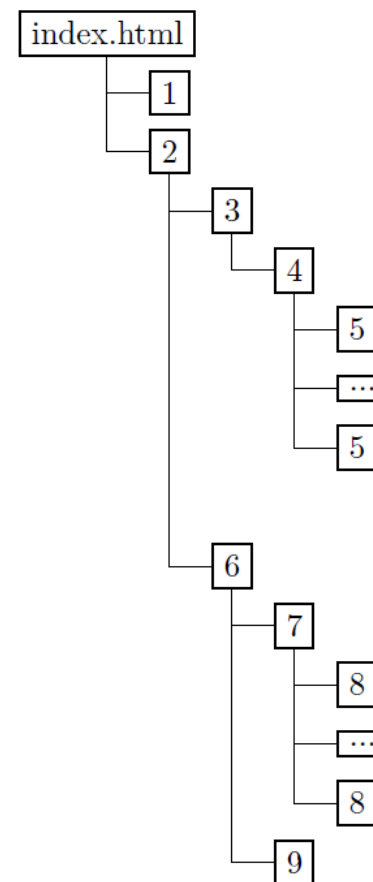


Kreiranje hijerarhije komponenti

Podela aplikacije na komponente

Primer 2

1. Zaglavlje stranice
2. Glavni deo
3. Upravljanje kursevima
4. Lista kurseva
5. Odabir kursa
6. Upravljanje informacijama o kursu
7. Lista opcija
8. Odabir opcije
9. Detalji opcije



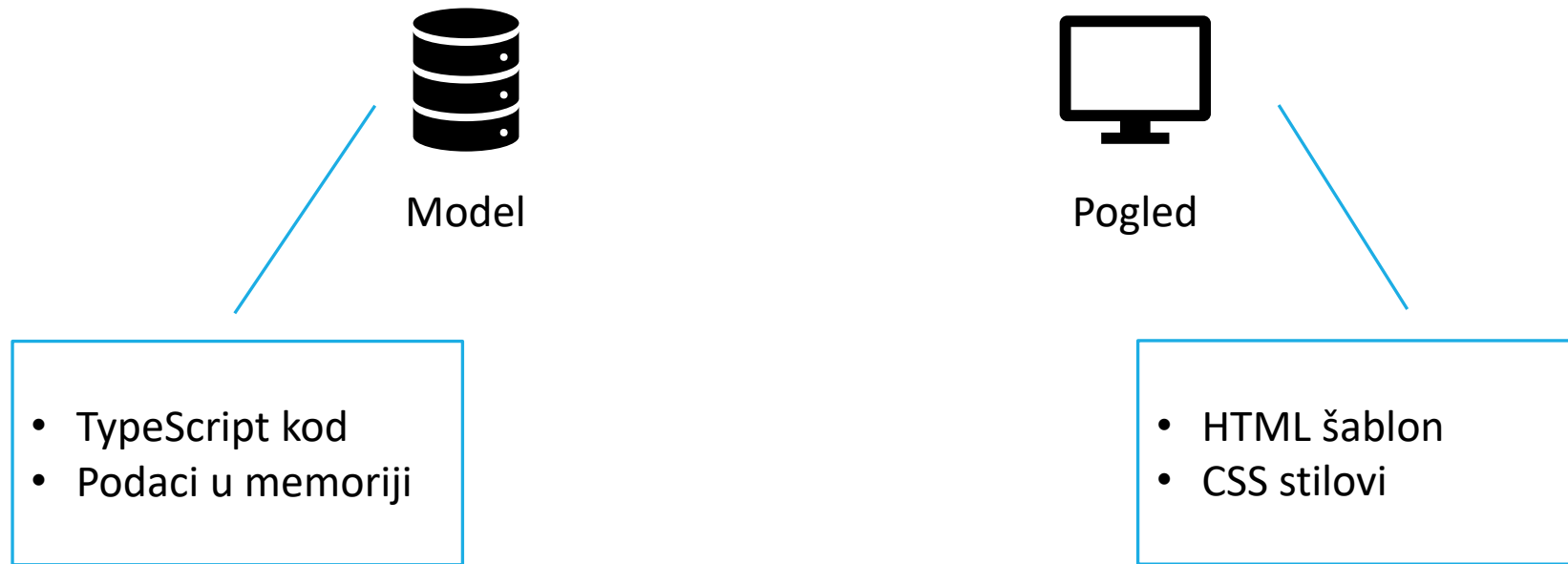
Angular CLI – kreiranje komponente

`ng generate component naziv-komponente`

- `src/app/naziv-komponente/naziv-komponente.html`
 - HTML šablon komponente
- `src/app/naziv-komponente/naziv-komponente.css`
 - CSS stilovi koji će biti primenjeni samo na elemente iz HTML šablona komponente
- `src/app/naziv-komponente/naziv-komponente.ts`
 - TypeScript kod koji definiše ponašanje i osobine komponente
- `src/app/naziv-komponente/naziv-komponente.spec.ts`
 - TypeScript kod koji testira ponašanje komponente
 - Koristi Jasmine biblioteku za pisanje testova (<https://jasmine.github.io/>)

Vezivanje podataka

Komponente imaju dva dela



Vezivanje podataka

Vezivanje podataka predstavlja

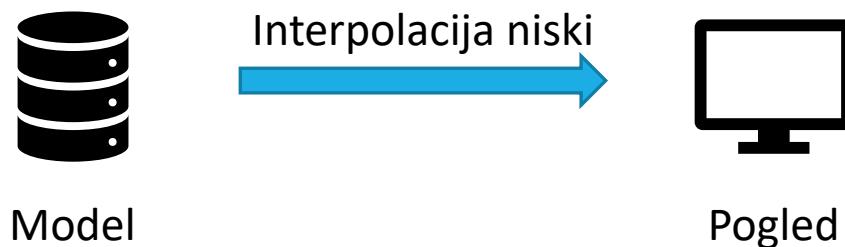
- Mehanizam kojim se dinamički podaci koji se kreiraju izvršavanjem koda u modelu koriste za prikazivanje komponente u pogledu
- Mehanizam kojim izmene u pogledu utiču na podatke u modelu

Nivoi vezivanja podataka

- Usmerenost
 - U smeru od modela ka pogledu – izmene u podacima utiču na prikaz šablona
 - U smeru od pogleda ka modelu – izmene u šablonu utiču na podatke
- Kardinalnost
 - U jednom smeru – izmene na jednoj strani utiču na drugu stranu, ali ne i obrnuto
 - U oba smera – promena bilo na jednoj bilo na drugoj strani utiče na suprotnu stranu

Interpolacija niski

Mehanizam kojim se podaci koji se nalaze u modelu izračunavaju i prikazuju u pogledu



Sintaksa

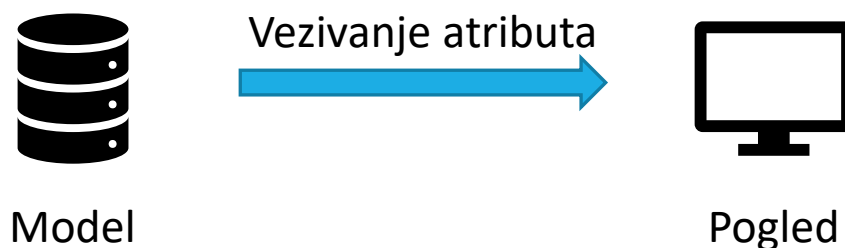
- U HTML šablonu se navode dvostruke vitičaste zagrade između kojih se nalazi *šablonski izraz*
- Šablonski izraz se izračunava, konvertuje u nisku i upisuje se umesto zagrada u šablonu
- Primeri:

```
<h1>{{ this.imePrezime }}</h1>
```

```
<p>Moja email adresa je: {{ dohvatiEmailAdresu() }}</p>
```

Vezivanje atributa

Mehanizam kojim se podaci koji se nalaze u modelu koriste za definisanje vrednosti HTML atributa elemenata u pogledu



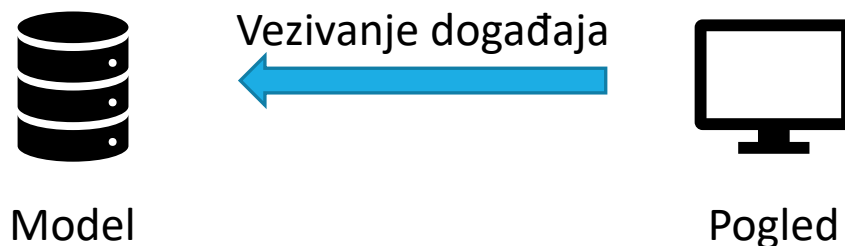
Sintaksa

- U HTML šablonu se navode uglaste zagrade između kojih se nalazi *naziv HTML atributa*
- Vrednost je *šablonski izraz* čija vrednost će biti postavljena za vrednost tog HTML atributa
- Primer:

```
<input [required]="true">
```

Vezivanje događaja

Mehanizam kojim se reaguje na promene u pogledu radi ažuriranja modela



Sintaksa

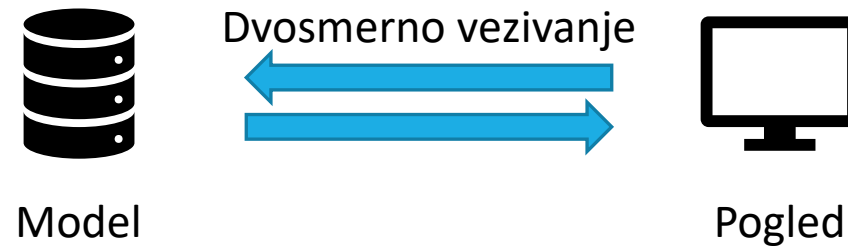
- U HTML šablonu se navode obične zagrade između kojih se nalazi *naziv HTML događaja*
- Vrednost je *poziv metoda komponente* koji će biti izvršen pri ispaljivanju tog HTML događaja nad tim HTML elementom
- Primeri:

```
<input (click)="onClick()">
```

```
<input (input)="onInputChangeValue($event)">
```

Dvosmerno vezivanje

Mehanizam koji implementira promenu vrednosti u oba smera



Sintaksa

- U HTML šablonu se navode i obične zagrade i uglaste zagrade između kojih se nalazi Angular atribut `ngModel`
- Vrednost je *naziv podatka u komponenti* čija će vrednost biti dvosmerno vezana za vrednost HTML elementa za koji se vrši dvosmerno vezivanje
- Primeri:

```
<input [(ngModel)]="imePrezime">
```

Šablonski izrazi

Jednolinijski TypeScript izrazi koji se navode prilikom vezivanja podataka

Smernice

- Bez vidljivih bočnih efekata
 - Ne sme menjati bilo koje stanje aplikacije osim vrednosti ciljanog svojstva
- Brza izračunljivost
 - Angular izvršava šablonske izraze nakon svakog ciklusa detekcije promena
 - Spori šablonski izrazi dovode do usporavanja aplikacije
 - Skuplja izračunavanja bi trebalo keširati
- Jednostavnost
 - Kompleksna poslovna logika bi trebalo da bude implementirana u modelu, a ne u šablonu

Hvala na pažnji!

NIKOLA AJZENHAMER

PROGRAMIRANJE ZA VEB

MATEMATIČKI FAKULTET

2020/2021