

Krátce o projektu

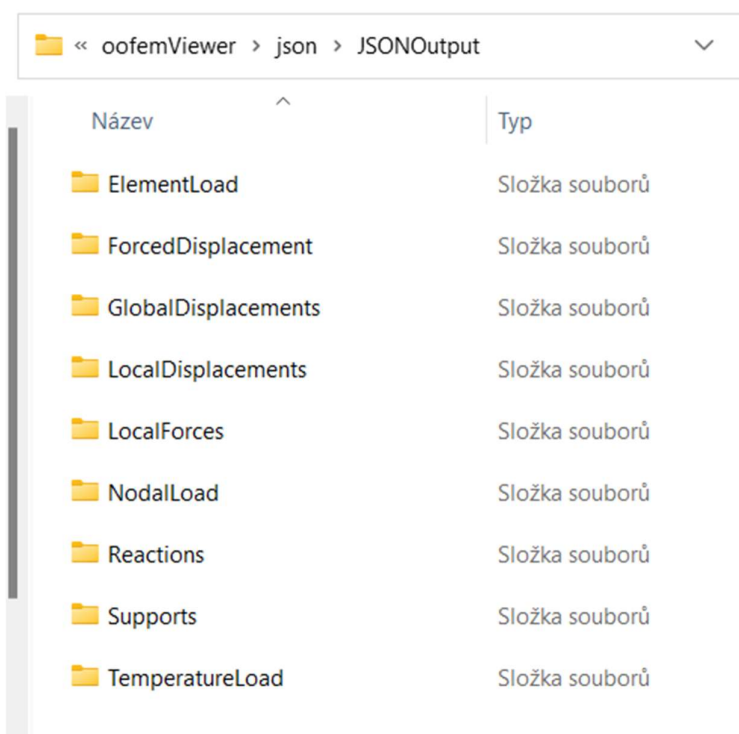
1.	Stažení aplikace.....	2
2.	Jak mne napadlo programovat?.....	3
3.	Co aplikace dělá?.....	3
4.	Jak aplikace funguje.....	3

1. Stažení aplikace

Projekt si můžeme stáhnout na adrese:

<https://github.com/jonasRower/oofemViewer>

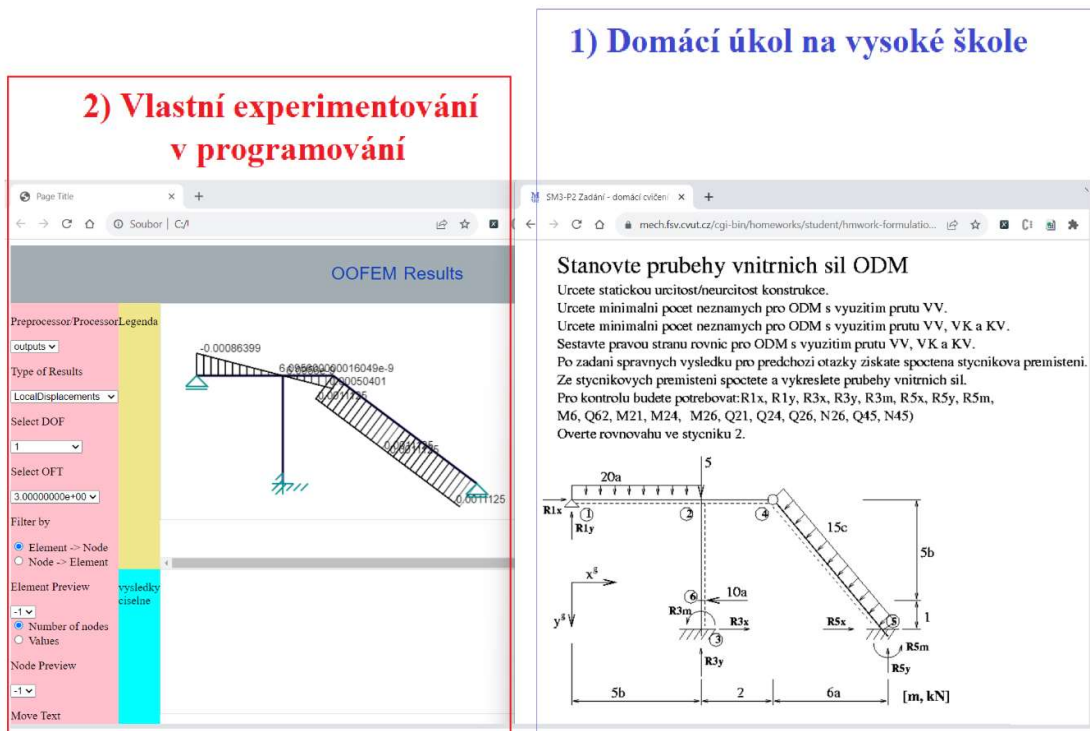
Projekt obsahuje stromovou strukturu s řady jsonů. Zdrojové soubory jsou rozřizené do adresářové struktury. Viz např. obrázek zde:



Obr. 1 – Složky ve kterých jsou uložena data

2. Jak mne napadlo programovat?

Programování mne napadlo již na Vysoké škole.



Obr. 2 – Počáteční motivace k programování.

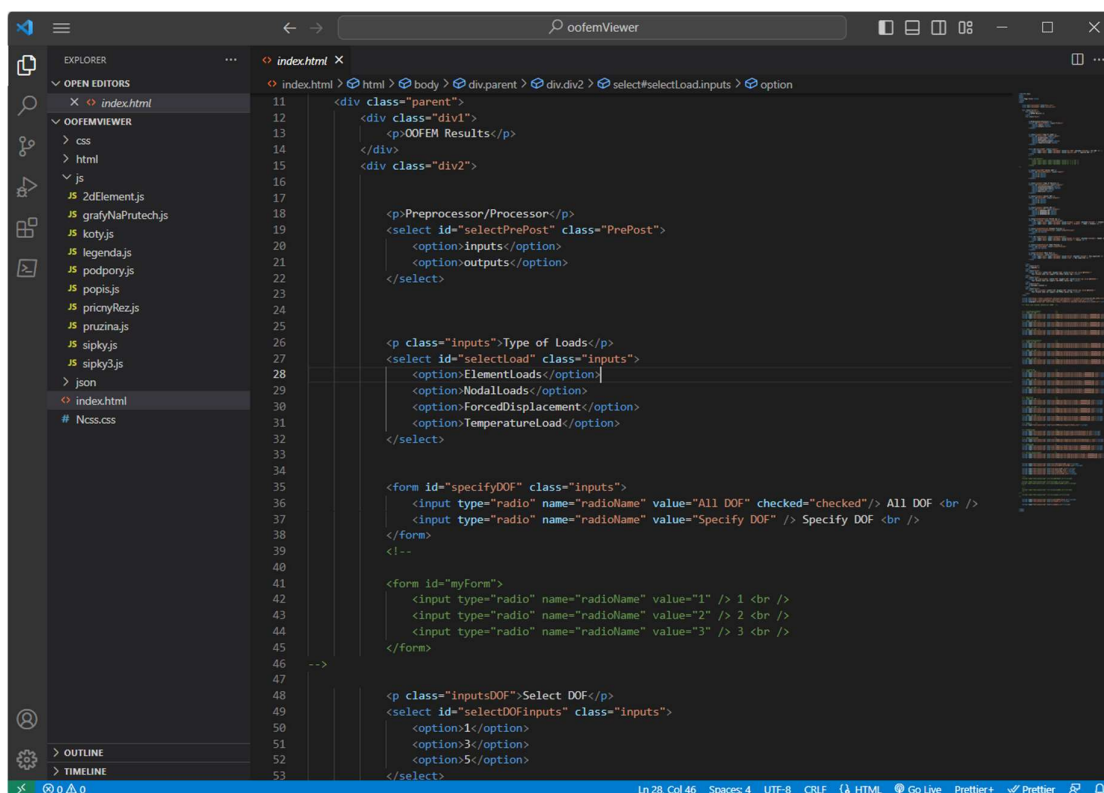
3. Co aplikace dělá?

Aplikace zobrazuje výsledky výpočtů. Není tedy třeba žádné databáze. Aplikace běží lokálně.

4. Jak aplikace funguje.

Aplikace obsahuje v levém sloupci panel s výběrem požadovaných hodnot. Výsledky se vykreslují pomocí Canvasu v grafice.

Kód aplikace – stručně.



```

11 <div class="parent">
12   <div class="div1">
13     <p>OOFEM Results</p>
14   </div>
15   <div class="div2">
16
17     <p>Preprocessor/Processor</p>
18     <select id="selectPrePost" class="PrePost">
19       <option>inputs</option>
20       <option>outputs</option>
21     </select>
22
23     <p class="inputs">Type of Loads</p>
24     <select id="selectLoad" class="inputs">
25       <option>ElementLoads</option>
26       <option>NodalLoads</option>
27       <option>ForcedDisplacement</option>
28       <option>TemperatureLoad</option>
29     </select>
30
31     <form id="specifyDOF" class="inputs">
32       <input type="radio" name="radioName" value="All DOF" checked="checked" /> All DOF <br />
33       <input type="radio" name="radioName" value="Specify DOF" /> Specify DOF <br />
34     </form>
35
36     <!--
37
38     <form id="myForm">
39       <input type="radio" name="radioName" value="1" /> 1 <br />
40       <input type="radio" name="radioName" value="2" /> 2 <br />
41       <input type="radio" name="radioName" value="3" /> 3 <br />
42     </form>
43
44     <p class="inputsDOF">Select DOF</p>
45     <select id="selectDOFinputs" class="inputs">
46       <option>1</option>
47       <option>3</option>
48       <option>5</option>
49     </select>
50
51   </div>
52 </div>
53

```

Obr. 3 – Html soubor.

Kód v souboru 2dElement.js zajišťuje dopočet souřadnic pro vykreslování prutů.



Obr. 4 – Jedna čára = jeden prut.

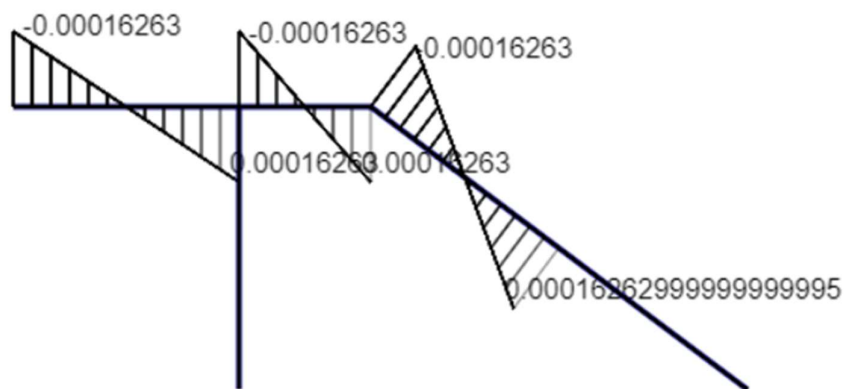
```

1
2
3 class elementy {
4
5   constructor(zvolenyJSONStr){
6
7       //ukládá všechna data, aby je mohl porovnávat se souřadnicí myši
8       this.vsechnySouradnice = [];
9       this.vsechnyCislaElementu = [];
10      this.vsechnaCislaVrcholu = [];
11      this.vsechnyBarvy = [];
12      this.vsechnyHodnotyVrcholu = [];
13      this.vsechnyHodnotyElementu = [];
14      this.vsechnyBarvyCarElementu = [];
15      this.vsechnytloustkyCarElementu = [];
16      this.zvolenyJSONStr = zvolenyJSONStr; //zvoleny JSON je momentalne string
17
18      //aby detekoval na jakem elementu se nachazi kurzor mysi, je potreba ziskat pole vseh usecek
19      this.poleVsechUsecek = [];
20      this.abPole = [];
21
22      //aby zobrazoval spravna cisla uzlu, je treba aby je seskupil do pole
23      this.poleVsechUzlu = [];
24
25      //tady se vybere adekvatni JSON, ten, který je zvolen comboboxem
26      this.elementy = eval(zvolenyJSONStr);
27
28      //this.elementy = xx;
29      this.vykresliVsechnyElementy();
30
31   }
32
33   //gettry:
34   getVsechnySouradnice(){
35       return(this.vsechnySouradnice);
36   }
37
38   getVsechnyCislaElementu(){
39       return(this.vsechnyCislaElementu);
40   }
41
42   getVsechnaCislaVrcholu(){
43

```

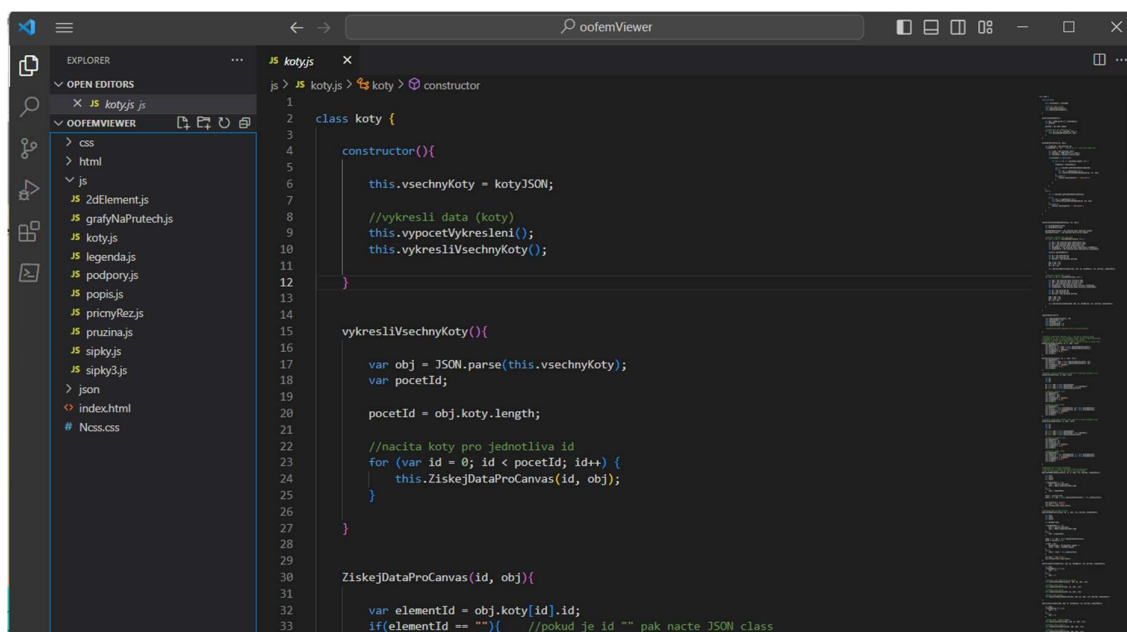
Obr. 5 – 2dElement.js

Kód v souboru grafyNaPrutech.js zajišťuje dopočet souřadnic pro hřebeny. Z obrázku je patrné, že je potřeba určitých matematických znalostí.



Obr. 6 – Grafy (neboli hřebeny) na prutech.

Kód původně zahrnoval i měření vzdáleností, to mělo probíhat pomocí souboru koty.js



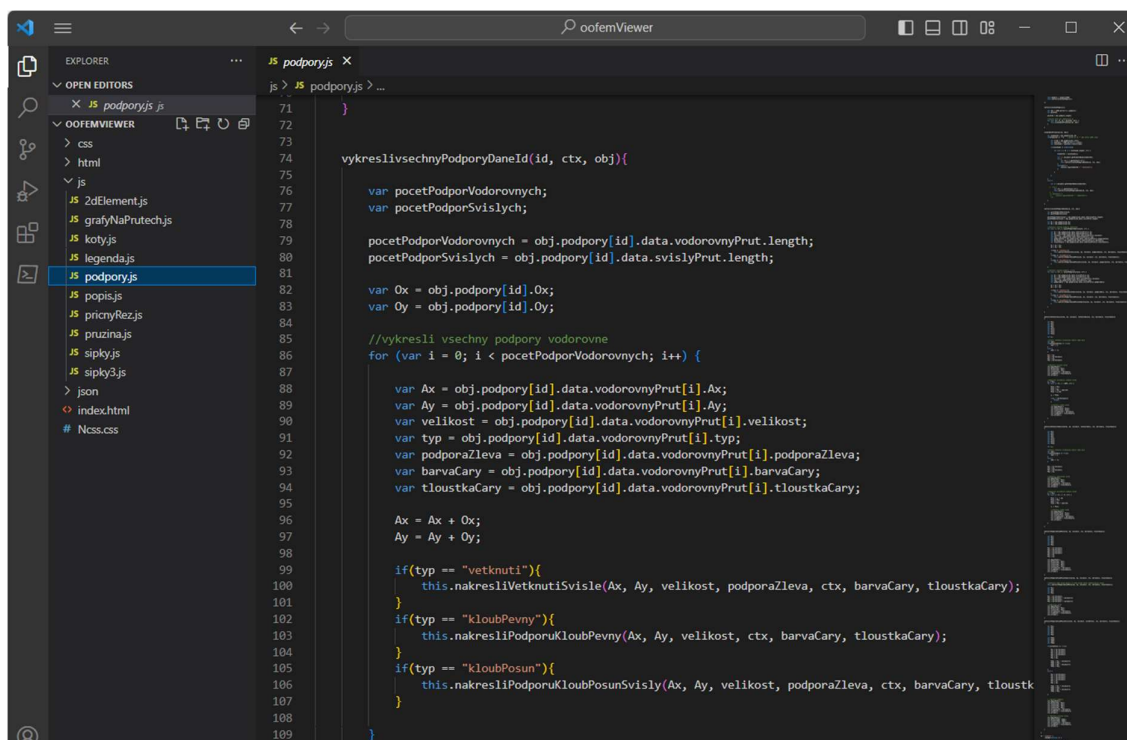
```

1  class koty {
2
3
4  constructor(){
5
6      this.vsechnyKoty = kotyJSON;
7
8      //vykresli data (koty)
9      this.vypocetVykresleni();
10     this.vykresliVsechnyKoty();
11 }
12
13
14
15
16 vykresliVsechnyKoty(){
17
18     var obj = JSON.parse(this.vsechnyKoty);
19     var pocetId;
20
21     pocetId = obj.koty.length;
22
23     //nacita koty pro jednotlivu id
24     for (var id = 0; id < pocetId; id++) {
25         this.ZiskejDataProCanvas(id, obj);
26     }
27 }
28
29
30 ZiskejDataProCanvas(id, obj){
31
32     var elementId = obj.koty[id].id;
33     if(elementId == ""){ //pokud je id "" pak nacte JSON class

```

Obr. 7 – koty.js

Kód souboru podpory.js zajišťuje vykreslování podpor.



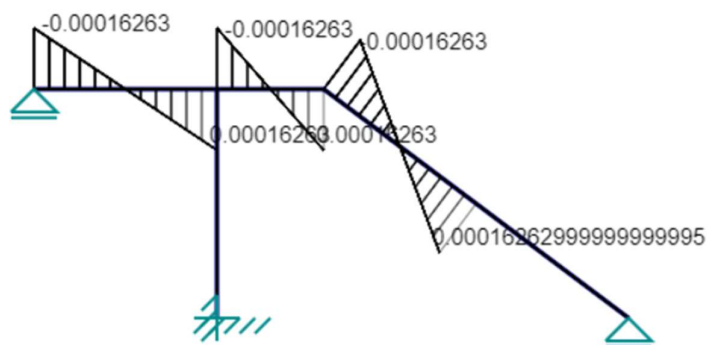
```

71 }
72
73
74 vykreslivsechnyPodporyDaneId(id, ctx, obj){
75
76     var pocetPodporVodorovnych;
77     var pocetPodporSvislych;
78
79     pocetPodporVodorovnych = obj.podpory[id].data.vodorovnyPrut.length;
80     pocetPodporSvislych = obj.podpory[id].data.svislyPrut.length;
81
82     var Ox = obj.podpory[id].Ox;
83     var Oy = obj.podpory[id].Oy;
84
85     //vykresli vsechny podpory vodorovne
86     for (var i = 0; i < pocetPodporVodorovnych; i++) {
87
88         var Ax = obj.podpory[id].data.vodorovnyPrut[i].Ax;
89         var Ay = obj.podpory[id].data.vodorovnyPrut[i].Ay;
90         var velikost = obj.podpory[id].data.vodorovnyPrut[i].velikost;
91         var typ = obj.podpory[id].data.vodorovnyPrut[i].typ;
92         var podporaZleva = obj.podpory[id].data.vodorovnyPrut[i].podporaZleva;
93         var barvaCary = obj.podpory[id].data.vodorovnyPrut[i].barvaCary;
94         var tloustkaCary = obj.podpory[id].data.vodorovnyPrut[i].tloustkaCary;
95
96         Ax = Ax + Ox;
97         Ay = Ay + Oy;
98
99         if(typ == "vetknuti"){
100             this.nakresliVetknutiSvisle(Ax, Ay, velikost, podporaZleva, ctx, barvaCary, tloustkaCary);
101         }
102         if(typ == "kloubPevny"){
103             this.nakresliPodporuKloubPevny(Ax, Ay, velikost, ctx, barvaCary, tloustkaCary);
104         }
105         if(typ == "kloubPosun"){
106             this.nakresliPodporuKloubPosunSvisly(Ax, Ay, velikost, podporaZleva, ctx, barvaCary, tloustkaCary);
107         }
108     }
109 }

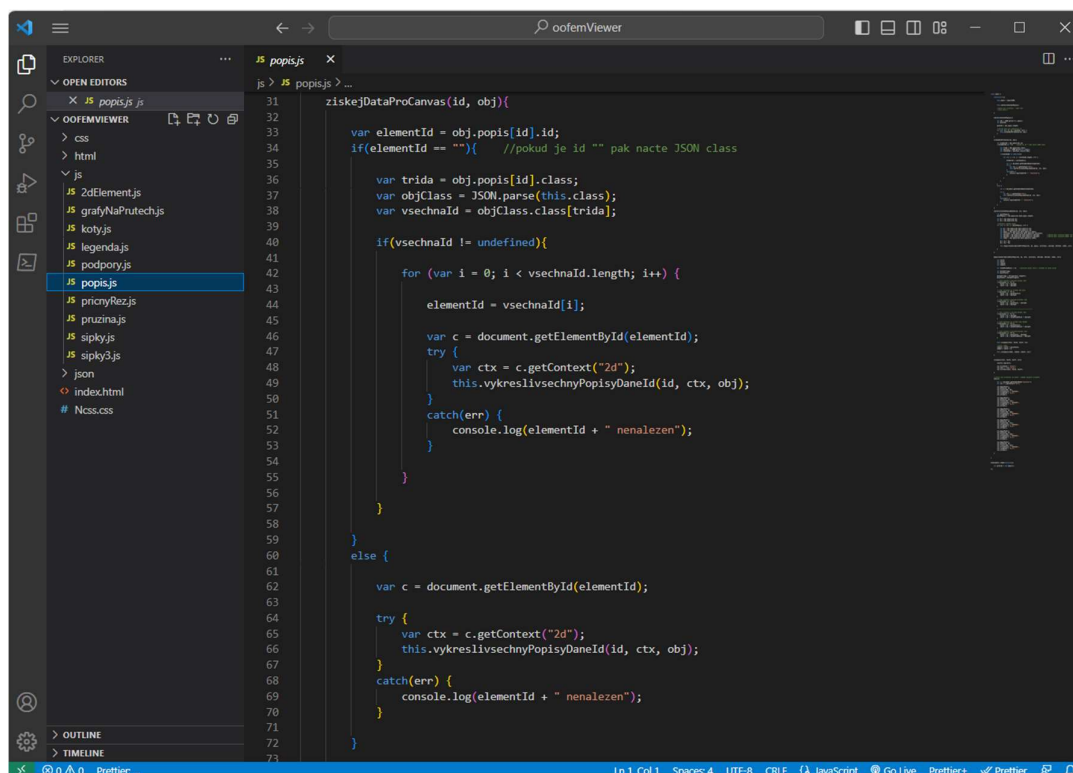
```

Obr. 8 – podpory.js

V souboru popis.js jsou dopočítávány souřadnice pro vykreslování popisů.

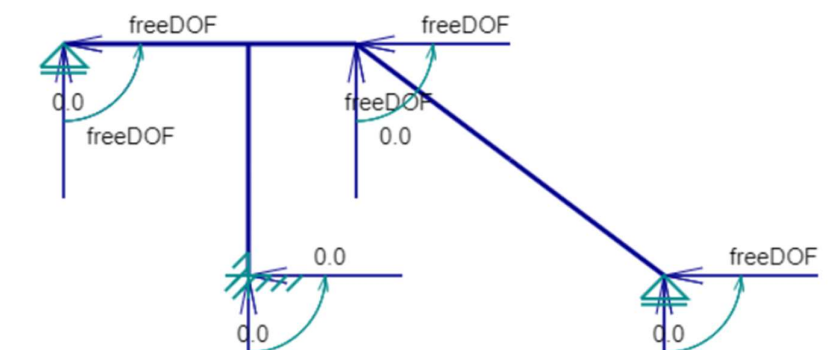


Obr. 10 – Popis na konstrukci.



Obr. 11 – Soubor popis.js

Soubor sipky.js zajišťuje vykreslování šipek.



Obr. 12 – Šipky na konstrukci

```

1  class sipky {
2
3
4
5
6
7
8      constructor(){
9
10         this.sipky = sipkyJSON;
11
12         //vykresli vsechny sipky
13         this.vykreslivsechnySipky()
14     }
15
16     vykreslivsechnySipky(){
17
18         var obj = JSON.parse(this.sipky);
19         var pocetId;
20
21         pocetId = obj.sipky.length;
22
23         //nacita koty pro jednotlivu id
24         for (var id = 0; id < pocetId; id++) {
25             this.ziskejDataProCanvas(id, obj);
26         }
27
28     }
29
30     ziskejDataProCanvas(id, obj){
31
32         try {
33             var elementId = obj.sipky[id].id;
34             var c = document.getElementById(elementId);
35             var ctx = c.getContext("2d");
36
37             this.vykreslivsechnySipkyDaneId(id, ctx, obj);
38         }
39         catch(err) {
40             console.log(elementId + " nenalezen");
41         }
42     }
43
44     ziskejDataProCanvas(id, obj){
45
46     }
47 }

```

Obr. 13 – sipky.js