

TEHNIČKA ŠKOLA RUĐERA BOŠKOVIĆA  
ZAGREB, GETALDIĆEVA 4

Niko Josipović

**APLIKACIJA ZA MORSEOVO KODIRANJE I DEKODIRANJE**

SEMINARSKI RAD

Zagreb, travanj 2025.

**SADRŽAJ**

<b>1</b>	<b>UVOD</b>	<b>1</b>
<b>2</b>	<b>PROGRAMI</b>	<b>2</b>
2.1	KODIRANJE . . . . .	2
2.2	DEKODIRANJE . . . . .	3
2.3	POKRETANJE . . . . .	4
2.4	PROMJENA FORMATA . . . . .	5

# 1 UVOD

Tema ovog seminarskog rada, izrađenog u sklopu predmeta *Uvod u računalne mreže*, jest razvoj aplikacije za kodiranje i dekodiranje Morseova koda. Cilj zadatka bio je implementirati softversko rješenje za pretvorbu teksta u Morseov kod i obrnuto, koristeći programski jezik po vlastitom izboru te dokumentirati proces izrade.

Aplikacija je razvijena korištenjem programskog jezika JavaScript, uz primjenu HTML-a i CSS-a za stvaranje sučelja i poboljšanje korisničkog iskustva. U nastavku će biti predstavljen i objašnjen kod koji omogućuje funkcionalnost aplikacije, zajedno s detaljima o njenoj implementaciji.

## 2 PROGRAMI

### 2.1 KODIRANJE

```
1 function encodeText(text) {
2   const words = text.toLowerCase().split(" ");
3
4   const encodedWords = words.map((word) => {
5     // Razdvajanje riječi na slova
6     const letters = Array.from(word);
7     const encodedLetters = letters.map((letter) => {
8       // Traženje Morseovog koda za svako slovo
9       return reverseDictionary[letter] || ''; // Ako slovo nije pronađeno↵
10      , koristi prazni string
11    });
12    return encodedLetters.join(" "); // Spajanje Morseovih simbola s ↵
13    jednim razmakom između njih
14  });
15
16  // Spajanje kodiranih riječi s trostrukim razmakom između riječi
17  let encodedText = encodedWords.join("  ");
18
19  const outputTextElement = document.getElementById('output-text');
20  outputTextElement.value = encodedText;
21
22  return encodedText;
23 }
```

**Program 1:** Funkcija koja pretvara običan teksta u Morseov kod

Funkcija *encodeText* implementira kodiranje teksta u Morseov kod. Sva slova ulaznog teksta su pretvorena u mala te je tekst razdvojen na riječi. Svaka riječ se dalje razdvaja na pojedinačna slova, koja se prevode u Morseove simbole korištenjem predefiniranog rječnika *reverseDictionary*. Ako slovo nije pronađeno u rječniku, vraća se prazan niz. Morseovi simboli unutar riječi spajaju se s jednim razmakom, dok se kodirane riječi odvajaju trostrukim razmakom. Konačni rezultat prikazuje se u HTML elementu s identifikatorom **output-text**.

## 2.2 DEKODIRANJE

```
1  function decodeMorseCode(morseCode) {
2      const words = morseCode.trim().split(" "); // Razdvajanje Morseovog ↵
        koda na riječi
3
4      const decodedWords = words.map((word) => {
5          const wordWithStandardSymbols = word.split("").map((symbol) => {
6              // Pretvorba Morseovog simbola u slovo
7              return dictionary[symbol] || ''; // Koristi prazni string ako ↵
                simbol nije pronađen u rječniku
8          }).join(""); // Spajanje znakova u riječ
9
10         return wordWithStandardSymbols;
11     });
12
13     const decodedText = decodedWords.join(" "); // Spajanje riječi u tekst
14
15     const outputTextElement = document.getElementById('output-text');
16     outputTextElement.value = decodedText; // Postavljanje dekodiranog ↵
        teksta u textarea
17
18     return decodedText;
19 }
```

**Program 2:** *Funkcija koja pretvara Morseov kod u običan tekst*

Funkcija *decodeMorseCode* dekodira Morseov kod u običan tekst. Ulazni Morseov kod uklanja suviše razmake metodom `trim()` i razdvaja na riječi koristeći trostruki razmak kao separator. Svaka riječ se dalje razdvaja na pojedinačne Morseove simbole jednostrukim razmakom. Simboli se prevode u slova korištenjem predefiniranog rječnika *dictionary*, pri čemu se za neprepoznate simbole vraća prazan niz. Dekodirana slova spajaju se u riječi, a riječi u konačni tekst s jednostrukim razmakom između njih. Rezultat se prikazuje u HTML elementu s identifikatorom **output-text**.

## 2.3 POKRETANJE

```
1 document.getElementById('input-text').addEventListener('input', function ↵
  () {
2   startTranslate();
3 });

1 function startTranslate() {
2   const inputTextElement = document.getElementById('input-text');
3   if (isInputLatin)
4   {
5     encodeText(inputTextElement.value);
6   } else
7   {
8     decodeMorseCode(inputTextElement.value);
9   }
10 }
```

**Program 3:** *Funkcije koje pokreću funkcije za kodiranje/dekodiranje*

Ovaj dio koda upravlja pokretanjem funkcija za kodiranje i dekodiranje Morseova koda. Oslušivač događaja (event listener) pridružen je HTML elementu s identifikatorom **input-text**, koji reagira na svaku promjenu unosa korisnika pozivom funkcije *startTranslate*.

Funkcija *startTranslate* provjerava format unosa putem varijable **isInputLatin**. Ako je *isInputLatin* postavljeno na *true*, unos je običan tekst i poziva se funkcija *encodeText* za pretvorbu u Morseov kod. U suprotnom, pretpostavlja se da je unos Morseov kod te se poziva funkcija *decodeMorseCode* za dekodiranje u običan tekst. Ovaj pristup omogućuje automatsku obradu unosa ovisno o njegovoj vrsti.

## 2.4 PROMJENA FORMATA

```
1  isInputLatin = 1;
2  function swapFunction() {
3      isInputLatin = 1 - isInputLatin; // I love this line of code
4
5      const inputElement = document.getElementById('input').querySelector('.↵
selected');
6      const outputElement = document.getElementById('output').querySelector('.↵
.selected');
7
8      const inputTextElement = document.getElementById('input-text');
9      const outputTextElement = document.getElementById('output-text');
10
11     const tempContent = inputElement.innerHTML;
12     const tempImageSrc = document.getElementById('input').querySelector('.↵
img').src;
13     const tempText = inputTextElement.value;
14
15     inputElement.innerHTML = outputElement.innerHTML;
16     document.getElementById('input').querySelector('img').src = document.↵
getElementById('output').querySelector('img').src;
17
18     outputElement.innerHTML = tempContent;
19     document.getElementById('output').querySelector('img').src = ↵
tempImageSrc;
20
21     inputTextElement.value = outputTextElement.value;
22     outputTextElement.value = tempText;
23
24     startTranslate();
25
26     const element = document.querySelector('.swap-position');
27     element.classList.toggle('rotated');
28 }
```

**Program 4:** *Funkcije koja se pokreće prilikom pritiska na gumb za promjenu formata unosa*

Funkcija *swapFunction* omogućuje zamjenu uloge polja za unos i izlaz, mijenjajući format obrade između latiničnog teksta i Morseova koda. Varijabla *isInputLatin* prebacuje se između 1 (latinični tekst) i 0 (Morseov kod) jednostavnom aritmetičkom operacijom. Zatim se zamjenjuju sadržaj, slike i tekst između polja za unos (input) i izlaz (output), uključujući pripadajuće HTML elemente i vrijednosti tekstualnih polja. Nakon zamjene, poziva se funkcija *startTranslate* kako bi se unos odmah preveo u novi format. Na kraju, klasa *rotated* se prebacuje na elementu *.swap-position*, čime se vizualno označava promjena - rotacijom ikone.