

## Vježbe 9

1.

a) Napisati funkciju koja računa  $n$ -ti Fibonacijev broj

b) Izmeriti vreme izvršavanja upotrebom sledećih naredbi:

```
const startTime = new Date();  
const endTime = new Date();  
const timeDifference = endTime.getTime() - startTime.getTime();
```

c) Uvesti memoizaciju u rešenje i ponovo izmeriti potrebno vreme

2.

Ispitajte da li su navedene funkcije čiste

a)

```
function pozdrav() {  
    console.log("Zdravo Pero");  
}
```

b)

```
function pozdrav() {  
    return "Zdravo Pero";  
}
```

b)

```
let ime = "Pera";  
function pozdrav() {  
    return "Zdravo" + ime;  
}
```

c)

```
function pozdrav() {  
    let ime = "Pera";  
    return "Zdravo" + ime;  
}
```

d)

```
function pozdrav(ime) {  
    return "Zdravo" + ime;  
}  
pozdrav("Pera");
```

e)

```
function pozdrav(ime) {  
    return "Zdravo" + ime;  
}  
let i = 'Pera';  
pozdrav(i);
```

3. Izmeniti funkciju *dodajBroj* tako da ona bude čista:

```
let niz = [8, 3, 1, 0];  
let broj = 6;  
function dodajBroj() {  
    niz.push(broj);  
    console.log(niz);  
}
```

*“Funktor je običan objekat koji implementira funkciju map koja se izvršava and svakom **vrednošću u objektu** da bi proizvela **novi objekat**.”*

*“Možemo ga tumačiti kao **kontejner** sa map operacijom koja omogućuje da se funkcija prosleđena map operaciji primeni na vrednosti u kontejneru.”*

4. Implementirati funktor *box* koji skladišti numeričku vrijednost u sebi.
5. Implementirati funktor *maybe* koji u sebi skladišti ili validnu ili nedostajuću vrijednost.