



OOP

Primjer rješavanja zadatka iz OOP

Zadatak

1. Napišite klasu **Auto** koja ima polja:

- **marka** (tekst)
- **startx** (cijeli broj)

i svojstvo: **brzina** (cijeli broj)

Auto se može kretati samo desno od 0 do 500, a kreće se pomoću metode **idi()** koja uvećava **startx** za vrijednost **brzina**, a vraća *true* ako ne prelazi 500. U protivnom vraća *false*.

2. Svojstvo **brzina** ograničava unos. Ako se pokuša postaviti vrijednost manja od nule, vrijednost polja brzina mora biti 2.
3. **Auto** sadrži metodu **razlika()** koja prima parametar tipa **Auto**, a vraća udaljenost dvije instance prema **startx**.
4. Napraviti dvije instance tipa **Auto** koji kreću s iste pozicije 0.
5. U datoteci **testiranje.js** upisati vrijednosti polja marka i brzina za svaki objekt. Objekti se kreću u desno pomoću metode **idi()**. Ispisati tko je pobijedio (marka) i za koliko točkica (pomoću metode **razlika()**).

Klasa i svojstva

- Napišite klasu **Auto** koja ima polja:
 - marka (tekst)
 - startx (broj)
- Svojstvo:
 - brzina (broj)

Polja

- Vježbe 02

```
constructor() {  
    this.marka = "";  
    this._brzina = 0;  
    this.startx = 0;  
}
```

idi()

- Metode klase (Vježbe 03, ...)

```
idi() {  
    this.startx += this.brzina;  
  
    if (this.startx >= 500) {  
        return false;  
    }  
    else {  
        return true;  
    }  
}
```

razlika()

- Prima parametar
- **Math.abs()** vraća apsolutnu vrijednost kako se ne bi dogodila negativna razlika
- Zadatak se može riješiti bez ovakvih dodatnih “ugrađenih” funkcija, ali ih nije zabranjeno koristiti ako ih znate

```
razlika(drugi) {  
    let r = Math.abs(this.startx - drugi.startx);  
    return r;  
}
```

Svojstvo brzina

- Svojstvo brzina ograničava unos
- Vježbe 04

```
get brzina() {  
    return this._brzina;  
}
```

```
set brzina(b) {  
    if (b < 0) {  
        this._brzina = 2;  
    }  
    else {  
        this._brzina = b;  
    }  
}
```

Stvaranje objekata

- Vježbe 02

```
let a = new Auto();  
let b = new Auto();
```


Unos podataka

- Ne možemo unositi podatke s tipkovnice kao što to radimo u C#-u
- Iz tog razloga ćemo testirati program upisivanjem vrijednosti

```
a.marka = "Ford";
```

```
a.brzina = 5;
```

```
b.marka = "Fiat";
```

```
b.brzina = -20;
```

Logika igre / obrada

- Utrka traje dok jedan od dva lika ne dođe do 500, a to je definirano u metodi
- Ovdje smo koristili **beskonačnu** petlju **while** (koja nas je malo podsjetila na petlju igre) u kombinaciji s **break** naredbom
 - To **nije najbolje rješenje** ovog problema jer odmah u prvom koraku radimo petlju koja će sigurno biti beskonačna, a takav program je teško prekinuti u *browseru* pogotovo u kombinaciji s **LiveServer** dodatkom
 - Pokušajte drugačiji pristup

```
while (true) {  
  
}
```



break

Logika igre / obrada

- Metoda **idi()** pomiče auto, ali također vraća true/false ovisno o tome je li se auto pomakao ili ne
- Ako je prešao 500 onda se smatra da je došao do cilja tako da mora vratiti false (više se ne miče)
- Ako se jedan od auta više ne miče, potrebno je ispisati tko je pobijedio
 - Marka
 - Razlika točkica u odnosu na drugog

Test podaci

- Za provjeru ćete imati testne podatke koje unosimo te podatke koji se moraju ispisati (točno kako piše)
- Stvarna rješenja će se testirati s više testnih podataka za sve moguće situacije

```
a.marka = "Ford";  
a.brzina = 5;
```

```
b.marka = "Fiat";  
b.brzina = -20;
```

Ford

302