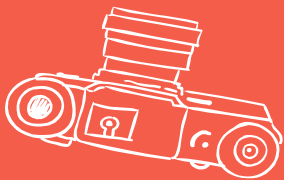


OBJEKTNO ORIJENTIRANO PROGRAMIRANJE






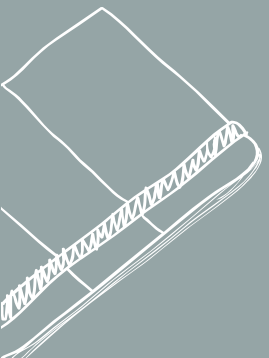

ČETIRI STUPA OBJEKTNO ORIJENTIRANOG PROGRAMIRANJA





POGLEDAJMO KAKO BI NAUČENO PRIMIJENILI

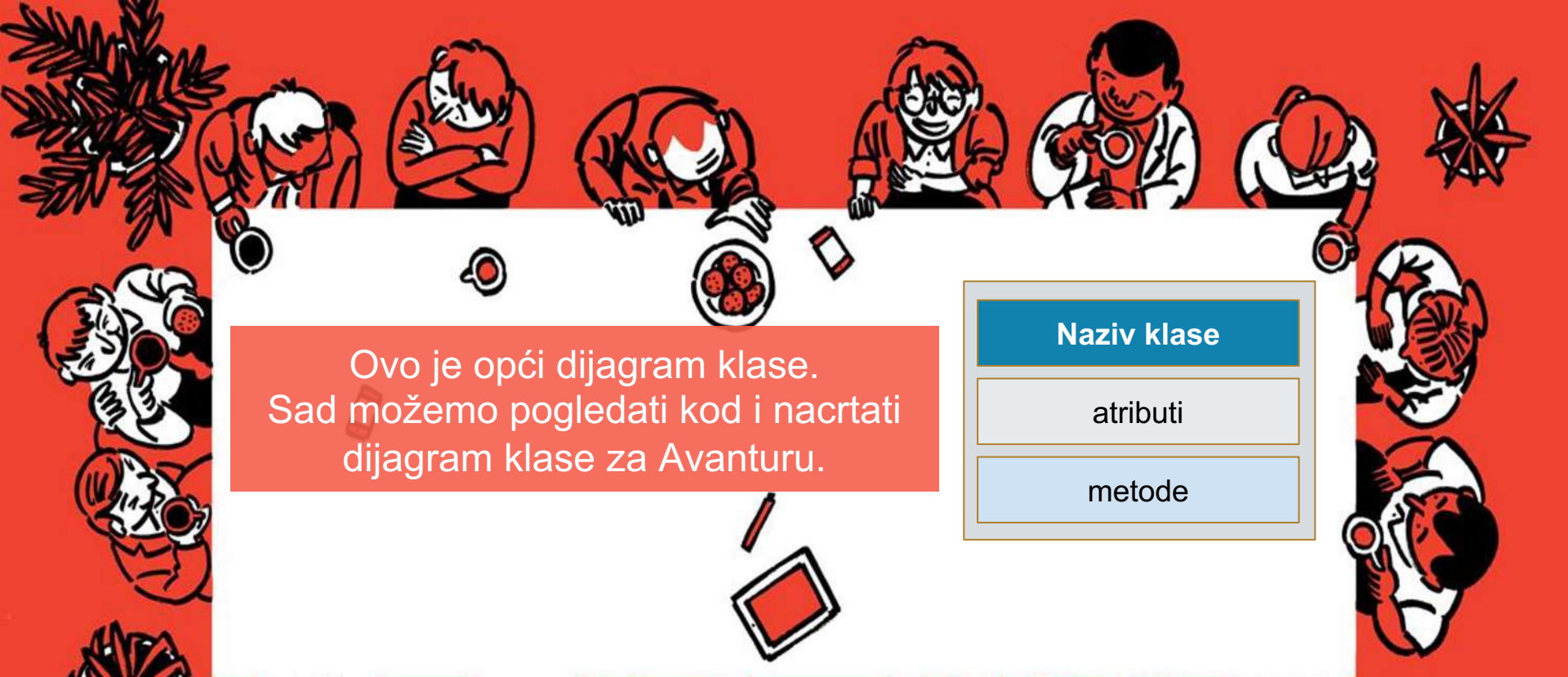
Započnimo s kodom naše igre Avantura, ali prije
toga oblikujmo (dizajnirajmo) aplikaciju!



Što radimo

Radimo dijagram klase (dio UML-a) koji je vrsta strukturnog dijagrama koji opisuje strukturu sustava objašnjavajući klase unutar sustava, njihove attribute i odnose.





Ovo je opći dijagram klase.
Sad možemo pogledati kod i nacrtati
dijagram klase za Avanturu.

Naziv klase

atributi

metode

Kako izgleda dijagram klase?

za trenutnu igru Avantura



Dijagram klasa

Kako izgleda dijagram klasa za igru koju smo počeli razvijati?

Naravno, zanima nas trenutni dijagram klasa.

Kod - Avantura

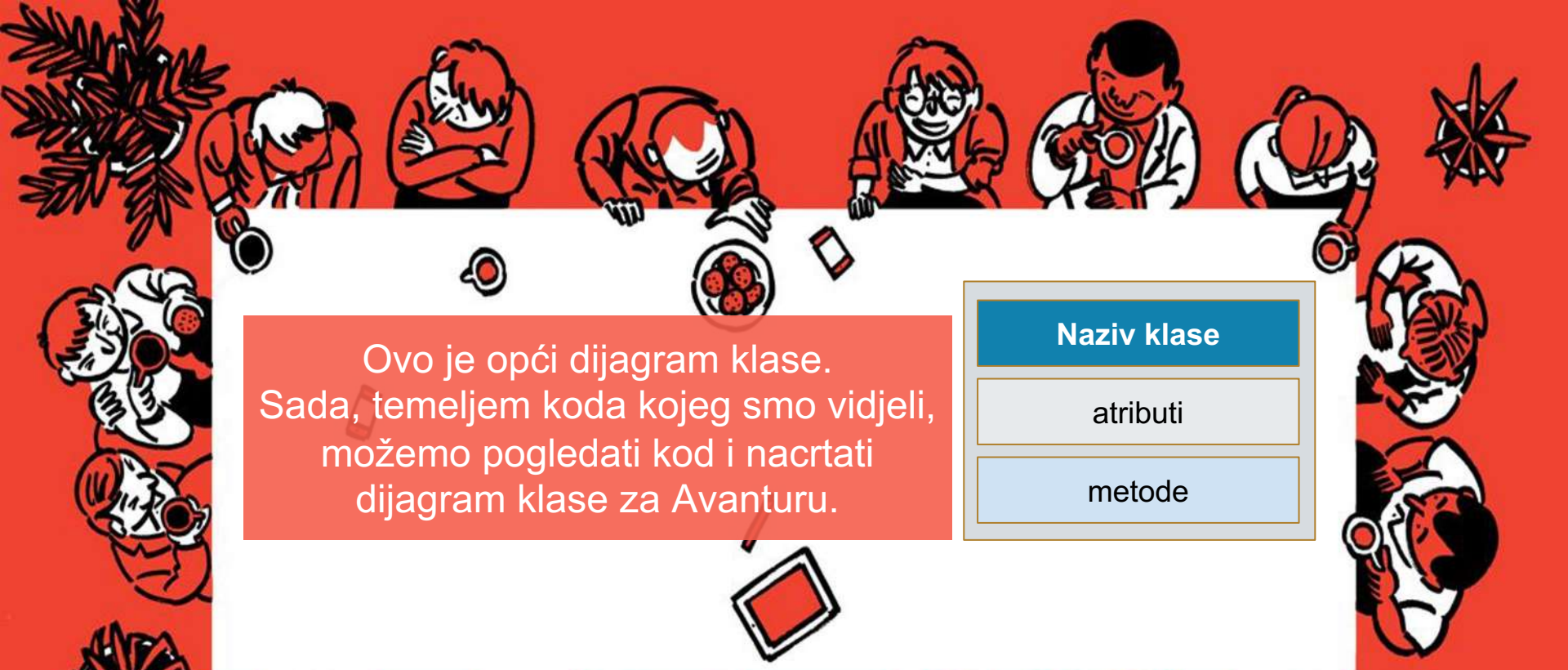


```
1 alert("Dobrodošli u Avanturu, igru OOP-a");  
2 console.log("Dobrodošli u Avanturu, igru OOP-a");  
3 let person = prompt("Kako želite nazvati vašeg lika?");  
4 alert("Odlično! Vaš lik se sada zove " + person);  
5 console.log("Odlično! Vaš lik se sada zove " + person);
```

Sve naredbe, do sada,
smještene su u index.js,
ulaznu točku u aplikaciju!

**Bilo bi dobro držati kod van
klasa na minimumu.**





Ovo je opći dijagram klase.
Sada, temeljem koda kojeg smo vidjeli,
možemo pogledati kod i nacrtati
dijagram klase za Avanturu.

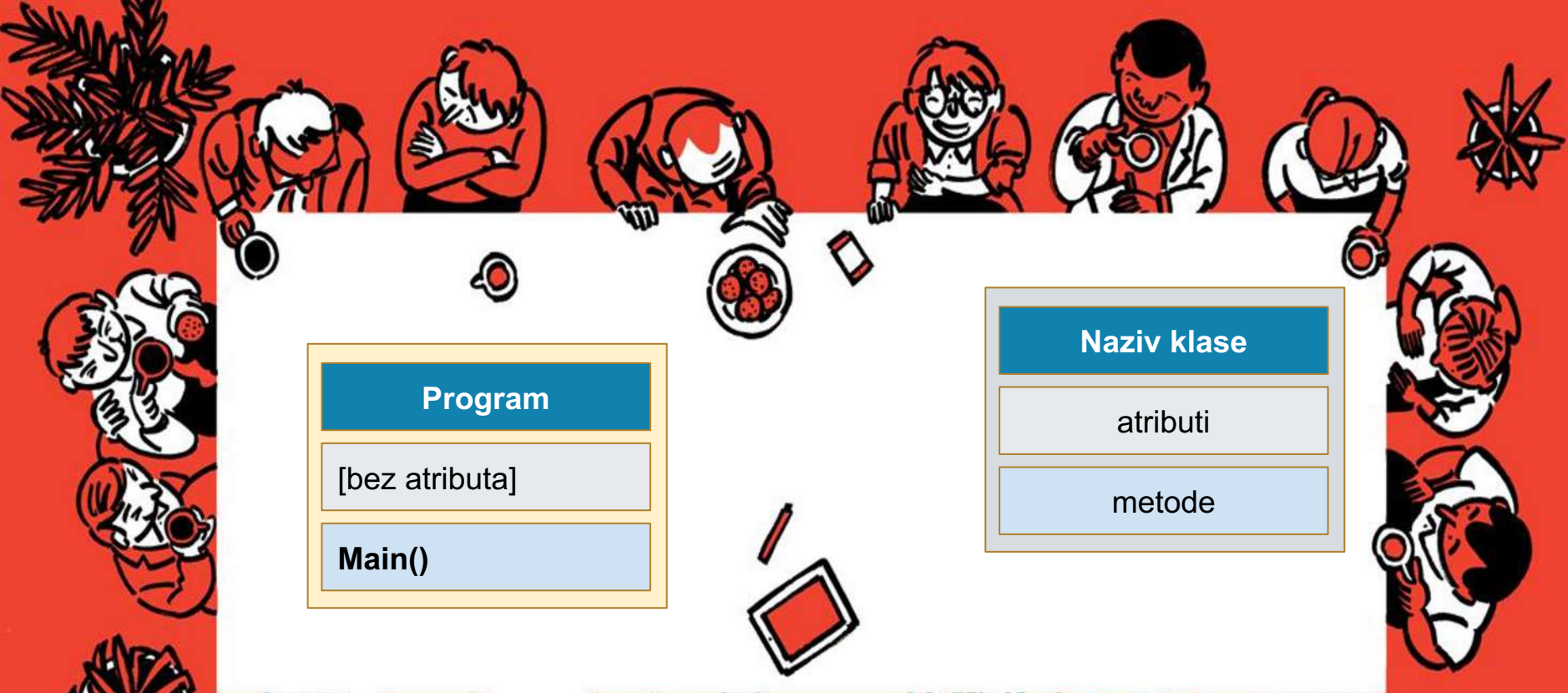
Naziv klase

atributi

metode

Kako izgleda dijagram klase?

za trenutnu igru Avantura



Kako izgleda dijagram klase?

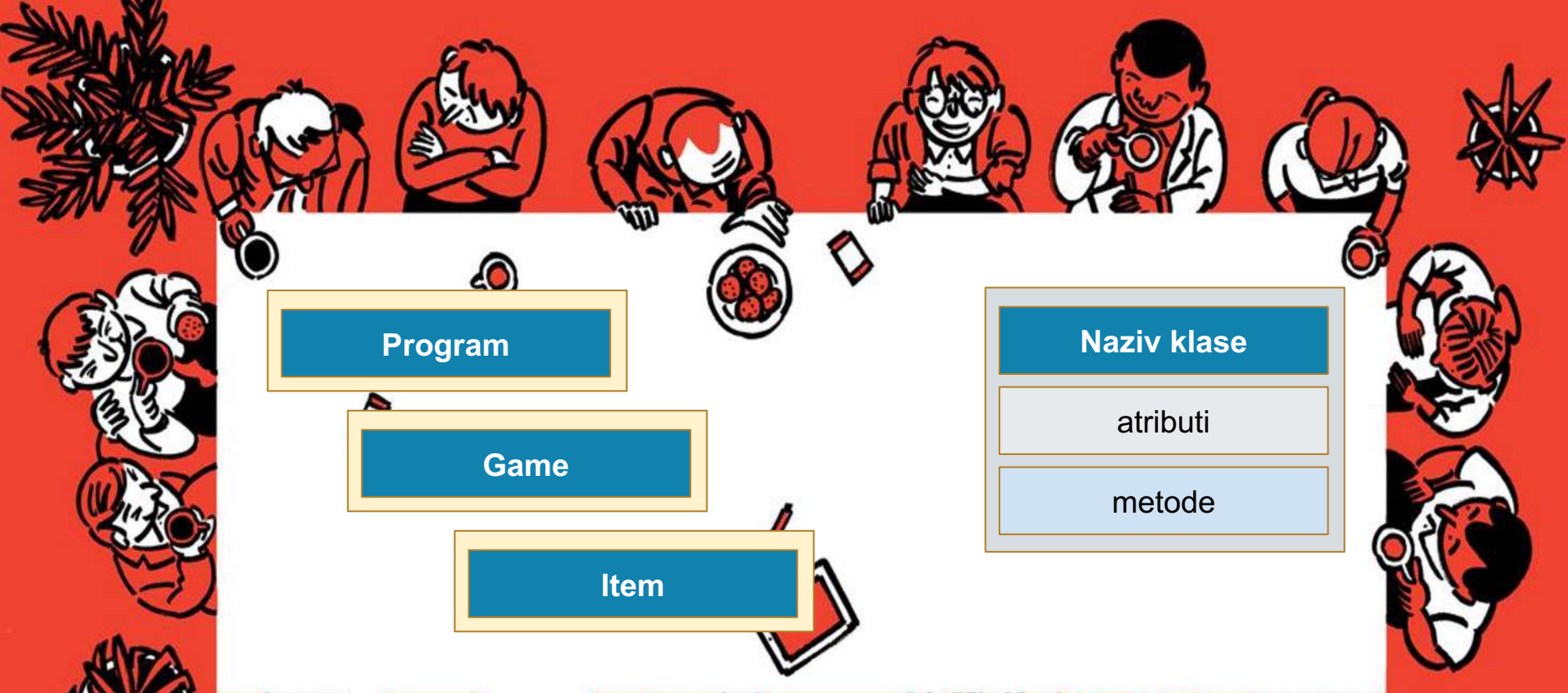
za trenutnu igru Avantura

Sastanak !



Način na koji ćete modelirati vašu klasu ovisi o konceptualizaciji aplikacije. Nema “pravog” načina izrade klasa.

Evo jednog primjera organizacije koda igre Avantura:



Program

Game

Item

Naziv klase

atributi

metode

Primjer organizacije koda

za trenutnu igru Avantura



Primjer organizacije koda

za trenutnu igru Avantura



Opća klasa koja će čuvati podatke o trenutno aktivnoj igri. Nećemo kreirati instance ove klase, pa je možemo izraditi kao statičku klasu.

Primjer organizacije koda



za trenutnu igru Avantura



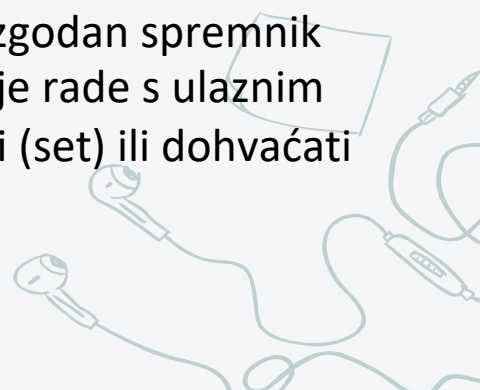
STATIČKA KLASA

Definicija - Statička klasa

Je klasa koja se ne može instancirati. Drukčije rečeno, nije dozvoljeno koristiti new operator kako bi se stvorila varijabla tipa class.



Statička klasa može se koristiti kao zgodan spremnik (container) za povezane metode koje rade s ulaznim parametrima i ne moraju postavljati (set) ili dohvaćati (get) interne podatke.



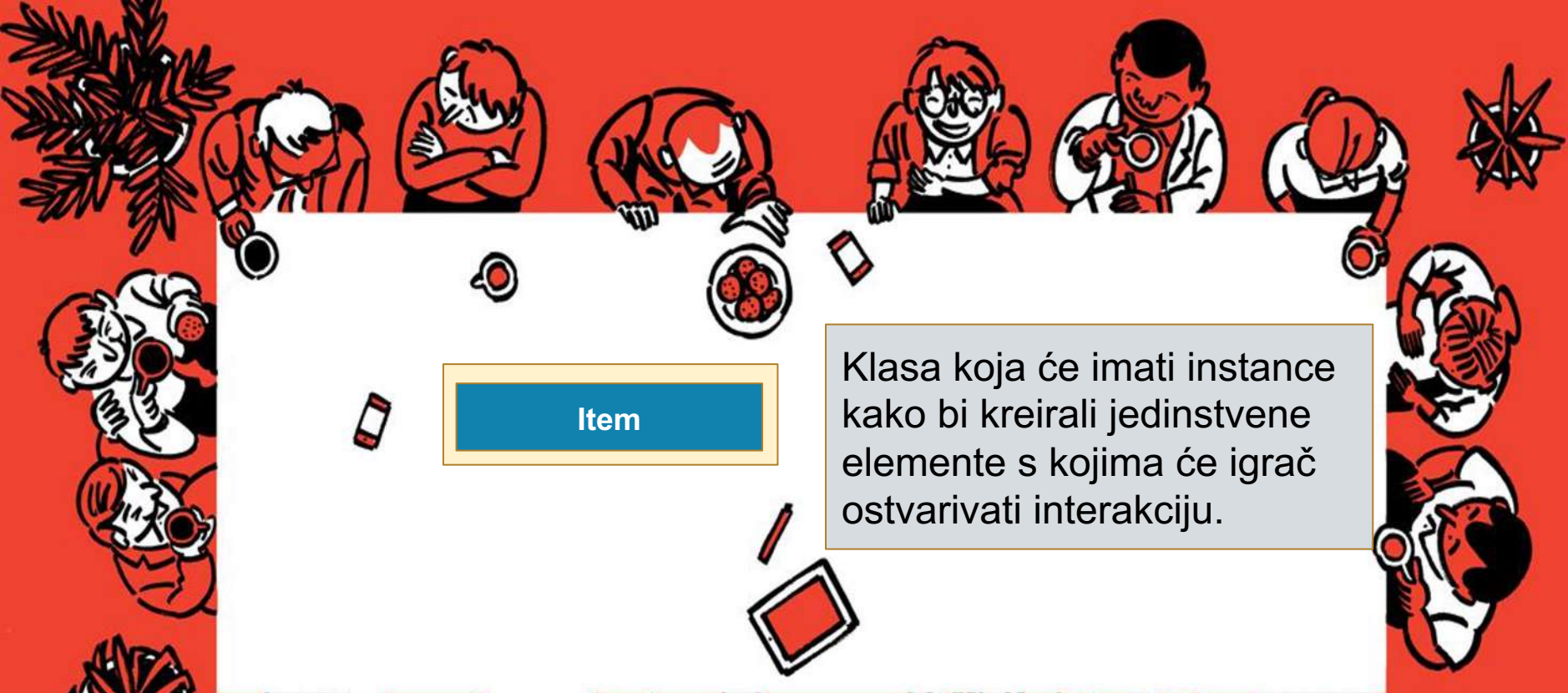
KAKO SPRIJEČITI INSTANCIRANJE KLASA?

Nemamo previše izbora, treba programirati

```
1  class StatickaKlasa {
2      constructor() {
3          if (this instanceof StatickaKlasa) {
4              throw Error("Statička klasa ne smije se instancirati.");
5          }
6      }
7      static method() {}
8  }
9
10 let o = new StatickaKlasa();
11 console.log(o);
```


KAKO RADI INSTANCEOF?

```
1  class Auto {  
2      constructor(proizvodjac, model, godina) {  
3          this.proizvodjac = proizvodjac;  
4          this.model = model;  
5          this.godina = godina;  
6      }  
7  }  
8  const auto = new Auto("VW", "Golf", 1974);  
9  
10 console.log(auto instanceof Auto);  
11 // očekivani izlaz: true
```



Klasa koja će imati instance kako bi kreirali jedinstvene elemente s kojima će igrač ostvarivati interakciju.

Primjer organizacije koda

za trenutnu igru Avantura



Primjer organizacije koda

za trenutnu igru Avantura

DO IDUĆI PUTA!

ste slobodni 