

Test broj 706

1. Funkcija u funkcionalnom programiranju treba da zadovoljava sledeće uslove (označite sva tvrđenja koja smatrate tačnim):

1. ☐ Da proizvodi rezultat samo na osnovu argumenata koje prima i da ne koristi ništa iz spoljašnjeg sveta.
2. ☐ Da proizvodi rezultat koristeći samo vrednosti iz spoljašnjeg sveta
3. ☐ Da proizvodi rezultat koristeći i svoje argumente i vrednosti iz spoljašnjeg sveta.

2. Označite sva tvrđenja koja smatrate tačnim:

1. ☐ PTC u JavaScript-u zahteva striktni režim.
2. ☐ PTC u JavaScript-u ne zahteva striktni režim.
3. ☐ PTC u JavaScript-u zahteva da se vrednost vrati eksplicitnom return naredbom.
4. ☐ PTC u JavaScript-u ne zahteva da se vrednost vrati eksplicitnom return naredbom.

3. Označite sva tvrđenja koja smatrate tačnim:

1. ☐ Osnovni problem pri korišćenju rekurzije je njena zahtevnost u pogledu memorijskih resursa (stek)
2. ☐ Osnovni problem pri korišćenju rekurzije je njena zahtevnost u pogledu procesorskih resursa.
3. ☐ Nema nikakvih posebnih problema u korišćenju rekurzije

4. Označite sva tvrđenja koja smatrate tačnim:

1. ☐ Funkcija map () omogućuje transformaciju putem transformacione funkcije koja joj se prosleđuje kao argument.
2. ☐ Transformaciona funkcija funkcije map () mora biti definisana u okviru funkcije map().
3. ☐ Funkcija map() je funkcija višeg reda.

5. Označite sva tvrđenja koja smatrate tačnim:

1. ☐ Map ugovor funktora Maybe je pogodna apstrakcija kojom može da se reši deo rukovanja greškama.
2. ☐ Map ugovor funktora Maybe je sve što je potrebno za potpunu apstrakciju za rukovanje greškama.
3. ☐ Map ugovor funktora Maybe ne može ni na koji način da se iskoristi kao apstrakcija za rukovanje greškama.

6. Primena funkcija višeg reda (označite sva tvrđenja koja smatrate tačnim):

1. ☐ Omogućuje da funkcije modifikuju druge funkcije
2. ☐ Ne omogućuje da funkcije modifikuju druge funkcije

7. Softverska komponenta (označite sva tvrđenja koja smatrate tačnim):

- ☐ Nema nikakvu ulogu u apstrakciji softvera.
- ☐ Predstavlja softversku jedinicu funkcionalnosti za upravljanje pojedinačnom apstrakcijom.
- ☐ Po nekad se koristi kao pomoćno sredstvo za softversku apstrakciju
- ☐ Predstavlja softversku jedinicu funkcionalnosti za upravljanje skupom raznorodnih apstrakcija.

8. Označite sva tvrđenja koja smatrate tačnim:

- ☐ Funkcija `map()` u jeziku JavaScript može kao ulaz da primi samo vrednosti primitivnog tipa.
- ☐ Funkcija `reduce()` u jeziku JavaScript može kao ulaz da primi samo vrednosti primitivnog tipa.
- ☐ Funkcija `filter()` u jeziku JavaScript može kao ulaz da primi samo vrednosti primitivnog tipa.
- ☐ Funkcije `map()`, `filter()` i `reduce()` mogu kao ulaz da prime vrednosti neprimitivnog tipa.

9. U jeziku JavaScript (označite sva tvrđenja koja smatrate tačnim):

- ☐ leksički doseg određen je izvornim kodom.
- ☐ u vreme izvršavanja koristi se samo čist leksički doseg baz ikakvog dopunjavanja.
- ☐ Leksički doseg se dinamički (u vreme izvršavanja) dopunjava evaluacijom sturukture koja sadrži varijable, funkcije i parametre definisane lokalno u okviru date funkcije.

10. 10. Označite sva tvrđenja koja smatrate tačnim:

- ☐ Pri kompoziciji $h(g(f(x)))$, tip izlaza funkcije f mora da odgovara tipu ulaza funkcije g .
- ☐ Pri kompoziciji $h(g(f(x)))$, tip izlaza funkcije f mora da odgovara tipu ulaza funkcije h .
- ☐ Pri kompoziciji $h(g(f(x)))$, tip izlaza funkcije g mora da odgovara tipu ulaza funkcije f .

11. U funkcionalnom programiranju (označite sva tvrđenja koja smatrate tačnim):

- ☐ Princip imutabilnosti zastupa nepromenljivost vrednosti.
- ☐ Princip imutabilnosti zastupa smanjenje potrošnje memorijskih resursa višekratnim korišćenjem identifikatora promenljivih.
- ☐ Princip imutabilnosti je fundamentalni princip.
- ☐ Princip imutabilnosti je sporedan princip.

12. Zatvaranje je mehanizam koji (označite sva tvrđenja koja smatrate tačnim):

- ☐ Omogućuje da unutrašnja funkcija pristupi dosegu svoje spoljašnje funkcije nakon što spoljašnja funkcija završi izvršavanje.
- ☐ Omogućuje da spoljašnja funkcija pristupi dosegu unutrašnje funkcije nakon što unutrašnja funkcija završi izvršavanje
- ☐ Omogućuje da funkcija pristupi običnom objektu koji je van nje deklarisan.

13. Operacija reduce (označite sva tvrđenja koja smatrate tačnim):

1. ☐ Kombinuje ("redukuje") listu na jednu vrednost (recimo, vrednost kao što su broj ili string)
2. ☐ Apstraktno se definiše kao uzimanje dve vrednosti i pravljenje jedne vrednosti od njih.
3. ☐ Apstraktno se definiše kao uzimanje jedne vrednosti i pravljenje dve ili više vrednosti od te jedne vrednosti.

14. Označite sva tvrđenja koja smatrate tačnim:

1. ☐ Memoizacija je optimizaciona tehnika za skladištenje prethodno izvršenih računanja.
2. ☐ Memoizacija je optimizaciona tehnika za efikasno kopiranje objekata.
3. ☐ Memoizacija je tehnika za implementaciju zatvaranja funkcije.

15. Označite sva tvrđenja koja smatrate tačnim

1. ☐ Operacija map uzima vrednost iz kontejnera, primenjuje primljenu funkciju transformacije na uzetu vrednost i rezultat vraća u kontejner.
2. ☐ Operacija map uzima vrednost iz kontejnera, primenjuje primljenu funkciju transformacije na uzetu vrednost, ali rezultat ne vraća u kontejner.
3. ☐ Operacija map može da uzme vrednost iz kontejnera i da tu tu vrednost nepromenjenu vrati u kontejner.

16. Parcijalna aplikacija je (označite sva tvrđenja koja smatrate tačnim):

1. ☐ Funkcija koja je primenjena na neke, ali ne na sve svoje argumente.
2. ☐ Funkcija koja je primenjena samo na svoj prvi argument.
3. ☐ Funkcija koja je primenjena samo na svoj poslednji argument.

17. U jeziku JavaScript (označite sva tvrđenja koja smatrate tačnim):

1. ☐ Redosled navođenja funkcija pri kompoziciji je obrnut od redosleda njihove primene.
2. ☐ Redosled navođenja funkcija pri kompoziciji odgovara redosledu njihove primene.

18. Point-free stil kodiranja je (označite sva tvrđenja koja smatrate tačnim):

1. ☐ stil gde se u definiciji funkcija ne navode parametri funkcije.
2. ☐ stil gde se u definiciji funkcija ne navodi ime funkcije.
3. ☐ stil koji jezik JavaScript ne omogućuje.

19. Osnovna ideja pri korišćenju funktora za rukovanje greškom je (označite sva tvrđenja koja smatrate tačnim)

1. ☐ Da se vrata različiti funktori u bloku try-catch.
2. ☐ Da se vrati jedan funktor koji u sebi sadrži svu logiku rukovanja greškom.

20. Označite sva tvrđenja koja smatrate tačnim:

- ☐ Operacija map() se može predstaviti i kao redukcija
- ☐ Ako se operacija map() predstavlja kao redukcija, inicijalna vrednost mora da bude prazna lista
- ☐ Operacija filter() ne može se predstaviti kao redukcija

Dat je sledeći kod:

```
const Container = function(val) {  
  this.value = val;  
}  
Container.of = function(value) {  
  return new Container(value);  
}  
Container.prototype.map = function(fn){  
  return Container.of(fn(this.value));  
}  
let double = (x) => x + x;  
let triple = (x) => x + x + x;  
Container.of(3).map(double)  
  .map(triple)  
  .map(double)
```

21.

- ☐ Kod koji je naveden nije sintaksno ispravan.
- ☐ Kod koji je naveden je sintaksno ispravan i kao rezultat vratiće Container {value: 36}
- ☐ Kod koji je naveden je sintaksno ispravan i kao rezultat vratiće Container {value: 18}

22. Označite sva tvrđenja koja smatrate tačnim:

- ☐ Komponovanje funktora je jednostavno jer su tipovi ulaza i izlaza usklađeni.
- ☐ Komponovanje funktora zahteva dodatnu transformaciju radi usklađivanja tipova ulaza i izlaza.
- ☐ Komponovanje funktora nije uopšte moguće.

23. Označite sva tvrđenja koja smatrate tačnim

- ☐ Funktor omogućuje ulančavanje funkcija koje vrše transformaciju podatak u kontejneru, ali je to ulančavanje veoma komplikovano
- ☐ Funktor ne omogućuje ulančavanje funkcija koje vrše transformaciju podatak u kontejneru
- ☐ Funktor omogućuje vrlo lako ulančavanje funkcija koje vrše transformaciju podatak u kontejneru

24. Funkcija višeg reda je funkcija koja (označite sva tvrđenja koja smatrate tačnim):

- ☐ Može da primi funkciju kao argument, ali ne može da vrati funkciju
- ☐ Može da vrati funkciju, ali ne može da primi funkciju kao argument
- ☐ Može i da vrati funkciju i da primi funkciju kao argument

Data su vam dva fragmenta koda

1. Fragment 1:

```
var x;  
if (y >= 0) {  
    x = y;  
} else {  
    x = -y;  
}
```

2. Fragment 2:

```
var x = y >= 0 ? y : -y;
```

Ta dva fragmenta se razlikuju (označite odgovore koje smatrate tačnim):

1. ☐ a. U rezultatu koji proizvode
2. ☐ b. U sintaksi
3. ☐ c. Ni u čemu se ne razlikuju

26. Spajanje lista je operacija koja (označite sva tvrđenja koja smatrate tačnim):

1. ☐ Formira rezultujuću listu preplitanjem odgovarajućih vrednosti iz dve liste.
2. ☐ Kombinuje elemente prve liste sa elementima druge liste tako što na njih primenjuje prosleđenu map() operaciju
3. ☐ Formira rezultujuću listu tako što iz prve liste eliminše elemente koji imaju iste vrednosti kao neki od elemenata druge liste

27. 25. Označite sva tvrđenja koja smatrate tačnim:

1. ☐ Signature tipa olakšavaju ljudima čitanje koda .
2. ☐ Signature tipa omogućuju mašinama da automatski zaključuju tipove u fazi izvršavanja.
3. ☐ Signature tipa ne omogućuju mašinama da automatski zaključuju tipove u fazi izvršavanja.

28. Označite sva tvrđenja koja smatrate tačnim:

1. ☐ Funktor omogućuje da se funkcija van konteksta primeni na vrednost u kontekstu.
2. ☐ Aplikativni funktor omogućuje da se funkcija u kontekstu primeni na vrednost u kontekstu.
3. ☐ Monada omogućuje mapiranje vrednosti a na b u situaciji gde je a u nekom kontekstu M(a) tako da se rezultat b dobija u kontekstu M(b).

29. Operacija pipe (označite sva tvrđenja koja smatrate tačnim):

1. ☐ Predstavlja način kompozicije funkcija
2. ☐ Predstavlja način kompozicije običnih objekata
3. ☐ Omogućuje kompoziciju sa-desna-na levo
4. ☐ Omogućuje kompoziciju sa-leva-na-desno

30. Zatvaranje je (označite sva tvrđenja koja smatrate tačnim):

1. ☐ Funkcija
2. ☐ Niz
3. ☐ logička varijabla

REŠENJA

Test broj 706

1. Funkcija u funkcionalnom programiranju treba da zadovoljava sledeće uslove (označite sva tvrđenja koja smatrate tačnim):

- ☒ Da proizvodi rezultat samo na osnovu argumenata koje prima i da ne koristi ništa iz spoljašnjeg sveta.
- ☐ Da proizvodi rezultat koristeći samo vrednosti iz spoljašnjeg sveta
- ☐ Da proizvodi rezultat koristeći i svoje argumente i vrednosti iz spoljašnjeg sveta.

2. Označite sva tvrđenja koja smatrate tačnim:

- ☒ PTC u JavaScript-u zahteva striktni režim.
- ☐ PTC u JavaScript-u ne zahteva striktni režim.
- ☒ PTC u JavaScript-u zahteva da se vrednost vrati eksplicitnom return naredbom.
- ☐ PTC u JavaScript-u ne zahteva da se vrednost vrati eksplicitnom return naredbom.

3. Označite sva tvrđenja koja smatrate tačnim:

- ☒ Osnovni problem pri korišćenju rekurzije je njena zahtevnost u pogledu memorijskih resursa (stek)
- ☐ Osnovni problem pri korišćenju rekurzije je njena zahtevnost u pogledu procesorskih resursa.
- ☐ Nema nikakvih posebnih problema u korišćenju rekurzije

4. Označite sva tvrđenja koja smatrate tačnim:

- ☒ Funkcija map () omogućuje transformaciju putem transformacione funkcije koja joj se prosleđuje kao argument.
- ☐ Transformaciona funkcija funkcije map () mora biti definisana u okviru funkcije map().
- ☒ Funkcija map() je funkcija višeg reda.

5. Označite sva tvrđenja koja smatrate tačnim:

- ☒ Map ugovor funktora Maybe je pogodna apstrakcija kojom može da se reši deo rukovanja greškama.
- ☐ Map ugovor funktora Maybe je sve što je potrebno za potpunu apstrakciju za rukovanje greškama.
- ☐ Map ugovor funktora Maybe ne može ni na koji način da se iskoristi kao apstrakcija za rukovanje greškama.

6. Primena funkcija višeg reda (označite sva tvrđenja koja smatrate tačnim):

- ☒ Omogućuje da funkcije modifikuju druge funkcije
- ☐ Ne omogućuje da funkcije modifikuju druge funkcije

7. Softverska komponenta (označite sva tvrđenja koja smatrate tačnim):

- ☐ Nema nikakvu ulogu u apstrakciji softvera.
- ☒ Predstavlja softversku jedinicu funkcionalnosti za upravljanje pojedinačnom apstrakcijom.
- ☐ Po nekad se koristi kao pomoćno sredstvo za softversku apstrakciju
- ☐ Predstavlja softversku jedinicu funkcionalnosti za upravljanje skupom raznorodnih apstrakcija.

8. Označite sva tvrđenja koja smatrate tačnim:

- ☐ Funkcija `map()` u jeziku JavaScript može kao ulaz da primi samo vrednosti primitivnog tipa.
- ☐ Funkcija `reduce()` u jeziku JavaScript može kao ulaz da primi samo vrednosti primitivnog tipa.
- ☐ Funkcija `filter()` u jeziku JavaScript može kao ulaz da primi samo vrednosti primitivnog tipa.
- ☒ Funkcije `map()`, `filter()` i `reduce()` mogu kao ulaz da prime vrednosti neprimitivnog tipa.

9. U jeziku JavaScript (označite sva tvrđenja koja smatrate tačnim):

- ☒ leksički doseg određen je izvornim kodom.
- ☐ u vreme izvršavanja koristi se samo čist leksički doseg baz ikakvog dopunjavanja.
- ☒ Leksički doseg se dinamički (u vreme izvršavanja) dopunjava evaluacijom sturukture koja sadrži varijable, funkcije i parametre definisane lokalno u okviru date funkcije.

10. 10. Označite sva tvrđenja koja smatrate tačnim:

- ☒ Pri kompoziciji $h(g(f(x)))$, tip izlaza funkcije f mora da odgovara tipu ulaza funkcije g .
- ☐ Pri kompoziciji $h(g(f(x)))$, tip izlaza funkcije f mora da odgovara tipu ulaza funkcije h .
- ☐ Pri kompoziciji $h(g(f(x)))$, tip izlaza funkcije g mora da odgovara tipu ulaza funkcije f .

11. U funkcionalnom programiranju (označite sva tvrđenja koja smatrate tačnim):

- ☒ Princip imutabilnosti zastupa nepromenljivost vrednosti.
- ☐ Princip imutabilnosti zastupa smanjenje potrošnje memorijskih resursa višekratnim korišćenjem identifikatora promenljivih.
- ☒ Princip imutabilnosti je fundamentalni princip.
- ☐ Princip imutabilnosti je sporedan princip.

12. Zatvaranje je mehanizam koji (označite sva tvrđenja koja smatrate tačnim):

- ☒ Omogućuje da unutrašnja funkcija pristupi dosegu svoje spoljašnje funkcije nakon što spoljašnja funkcija završi izvršavanje.
- ☐ Omogućuje da spoljašnja funkcija pristupi dosegu unutrašnje funkcije nakon što unutrašnja funkcija završi izvršavanje
- ☐ Omogućuje da funkcija pristupi običnom objektu koji je van nje deklarisan.

13. Operacija reduce (označite sva tvrđenja koja smatrate tačnim):

- ☒ 1. Kombinuje ("redukuje") listu na jednu vrednost (recimo, vrednost kao što su broj ili string)
- ☒ 2. Apstraktno se definiše kao uzimanje dve vrednosti i pravljenje jedne vrednosti od njih.
- ☐ 3. Apstraktno se definiše kao uzimanje jedne vrednosti i pravljenje dve ili više vrednosti od te jedne vrednosti.

14. Označite sva tvrđenja koja smatrate tačnim:

- ☒ 1. Memoizacija je optimizaciona tehnika za skladištenje prethodno izvršenih računanja.
- ☐ 2. Memoizacija je optimizaciona tehnika za efikasno kopiranje objekata.
- ☐ 3. Memoizacija je tehnika za implementaciju zatvaranja funkcije.

15. Označite sva tvrđenja koja smatrate tačnim

- ☒ 1. Operacija map uzima vrednost iz kontejnera, primenjuje primljenu funkciju transformacije na uzetu vrednost i rezultat vraća u kontejner.
- ☐ 2. Operacija map uzima vrednost iz kontejnera, primenjuje primljenu funkciju transformacije na uzetu vrednost, ali rezultat ne vraća u kontejner.
- ☒ 3. Operacija map može da uzme vrednost iz kontejnera i da tu tu vrednost nepromenjenu vrati u kontejner.

16. Parcijalna aplikacija je (označite sva tvrđenja koja smatrate tačnim):

- ☒ 1. Funkcija koja je primenjena na neke, ali ne na sve svoje argumente.
- ☐ 2. Funkcija koja je primenjena samo na svoj prvi argument.
- ☐ 3. Funkcija koja je primenjena samo na svoj poslednji argument.

17. U jeziku JavaScript (označite sva tvrđenja koja smatrate tačnim):

- ☒ 1. Redosled navođenja funkcija pri kompoziciji je obrnut od redosleda njihove primene.
- ☐ 2. Redosled navođenja funkcija pri kompoziciji odgovara redosledu njihove primene.

18. Point-free stil kodiranja je (označite sva tvrđenja koja smatrate tačnim):

- ☒ 1. stil gde se u definiciji funkcija ne navode parametri funkcije.
- ☐ 2. stil gde se u definiciji funkcija ne navodi ime funkcije.
- ☐ 3. stil koji jezik JavaScript ne omogućuje.

19. Osnovna ideja pri korišćenju funktora za rukovanje greškom je (označite sva tvrđenja koja smatrate tačnim)

- ☒ 1. Da se vrate različiti funktori u bloku try-catch.
- ☐ 2. Da se vrati jedan funktor koji u sebi sadrži svu logiku rukovanja greškom.

20. Označite sva tvrđenja koja smatrate tačnim:

- ☒ 1. Operacija map() se može predstaviti i kao redukcija
- ☒ 2. Ako se operacija map() predstavlja kao redukcija, inicijalna vrednost mora da bude prazna lista
- ☐ 3. Operacija filter() ne može se predstaviti kao redukcija

Dat je sledeći kod:

```
const Container = function(val) {  
  this.value = val;  
}  
Container.of = function(value) {  
  return new Container(value);  
}  
Container.prototype.map = function(fn){  
  return Container.of(fn(this.value));  
}  
let double = (x) => x + x;  
let triple = (x) => x + x + x;  
Container.of(3).map(double)  
  .map(triple)  
  .map(double)
```

21.

- ☐ 1. Kod koji je naveden nije sintaksno ispravan.
- ☒ 2. Kod koji je naveden je sintaksno ispravan i kao rezultat vratiće Container {value: 36}
- ☐ 3. Kod koji je naveden je sintaksno ispravan i kao rezultat vratiće Container {value: 18}

22. Označite sva tvrđenja koja smatrate tačnim:

- ☒ 1. Komponovanje funktora je jednostavno jer su tipovi ulaza i izlaza usklađeni.
- ☐ 2. Komponovanje funktora zahteva dodatnu transformaciju radi usklađivanja tipova ulaza i izlaza.
- ☐ 3. Komponovanje funktora nije uopšte moguće.

23. Označite sva tvrđenja koja smatrate tačnim

- ☐ 1. Funktor omogućuje ulančavanje funkcija koje vrše transformaciju podatak u kontejneru, ali je to ulančavanje veoma komplikovano
- ☐ 2. Funktor ne omogućuje ulančavanje funkcija koje vrše transformaciju podatak u kontejneru
- ☒ 3. Funktor omogućuje vrlo lako ulančavanje funkcija koje vrše transformaciju podatak u kontejneru

24. Funkcija višeg reda je funkcija koja (označite sva tvrđenja koja smatrate tačnim):

- ☐ 1. Može da primi funkciju kao argument, ali ne može da vrati funkciju
- ☐ 2. Može da vrati funkciju, ali ne može da primi funkciju kao argument
- ☒ 3. Može i da vrati funkciju i da primi funkciju kao argument

Data su vam dva fragmenta koda

1. Fragment 1:

```
var x;  
if (y >= 0) {  
    x = y;  
} else {  
    x = -y;  
}
```

2. Fragment 2:

```
var x = y >= 0 ? y : -y;
```

Ta dva fragmenta se razlikuju (označite odgovore koje smatrate tačnim):

- 1. ☐ a. U rezultatu koji proizvode
- 2. ☒ b. U sintaksi
- 3. ☐ c. Ni u čemu se ne razlikuju

26. Spajanje lista je operacija koja (označite sva tvrđenja koja smatrate tačnim):

- 1. ☒ Formira rezultujuću listu preplitanjem odgovarajućih vrednosti iz dve liste.
- 2. ☐ Kombinuje elemente prve liste sa elementima druge liste tako što na njih primenjuje prosleđenu map() operaciju
- 3. ☐ Formira rezultujuću listu tako što iz prve liste eliminše elemente koji imaju iste vrednosti kao neki od elemenata druge liste

27. 25. Označite sva tvrđenja koja smatrate tačnim:

- 1. ☒ Signature tipa olakšavaju ljudima čitanje koda .
- 2. ☒ Signature tipa omogućuju mašinama da automatski zaključuju tipove u fazi izvršavanja.
- 3. ☐ Signature tipa ne omogućuju mašinama da automatski zaključuju tipove u fazi izvršavanja.

28. Označite sva tvrđenja koja smatrate tačnim:

- 1. ☒ Funktor omogućuje da se funkcija van konteksta primeni na vrednost u kontekstu.
- 2. ☒ Aplikativni funktor omogućuje da se funkcija u kontekstu primeni na vrednost u kontekstu.
- 3. ☒ Monada omogućuje mapiranje vrednosti a na b u situaciji gde je a u nekom kontekstu M(a) tako da se rezultat b dobija u kontekstu M(b).

29. Operacija pipe (označite sva tvrđenja koja smatrate tačnim):

- 1. ☒ Predstavlja način kompozicije funkcija
- 2. ☐ Predstavlja način kompozicije običnih objekata
- 3. ☐ Omogućuje kompoziciju sa-desna-na levo
- 4. ☒ Omogućuje kompoziciju sa-leva-na-desno

30. Zatvaranje je (označite sva tvrđenja koja smatrate tačnim):

- 1. ☒ Funkcija
- 2. ☐ Niz
- 3. ☐ logička varijabla