

Također izjavljujem da mi zdravstveno stanje dozvoljava pisanje ove zadaće.

Vlastoručni potpis:

1. Općenito, za funkcionalnu jedinicu programskog sustava koja se naziva **komponenta** vrijedi da
 - A) mora imati vlastitu perzistenciju podataka
 - B) se nalazi u izoliranom procesu
 - C) obavlja isključivo pozive udaljenih procedura
 - D) ima javno sučelje opisano nekim od jezika za opis sučelja
 - E) se može neovisno postavljati i nadograđivati
2. Kod praktične primjene filozofije u kojoj se sva programska logika izmješta iz posredničkog sloja (*pipes*), sigurno se ne koristi:
 - A) arhitekturni stil REST
 - B) oblikovni obrazac *Tolerant Reader*
 - C) Enterprise Service Bus (ESB)
 - D) SOAP
 - E) asinkroni pozivi udaljenih procedura
3. U usporedbi s jediničnim (*unit*) testovima, testovi krajnjih točaka su
 - A) brži i obično ih ima više
 - B) sporiji i obično ih ima više
 - C) brži i obično ih ima manje
 - ☒ D) sporiji i obično ih ima manje
 - E) slični po brzini i opsegu
4. Ono što je WSDL u SOA tehnološkom stogu, to je za sustav gRPC
 - ☒ A) protocol buffers
 - B) Apache Thrift
 - C) UDDI
 - D) SOAP
 - E) XML

Glavna motivacija za korištenje sustava Apache Thrift je

- A) izolirano i udaljeno testiranje razvijenih komponenti
- B) ostvarenje objavi-pretplati raspodijeljene arhitekture koristeći RPC
- ☒ C) učinkovita i pouzdana komunikacija između komponenti ostvarenih primjenom različitih
- D) izbjegavanje vatrozida (*firewall*) korištenjem HTTP protokola
- E) potpora za objektno-orijentirani i proceduralni RPC

Prikazani isječak koda napisan je u jeziku za opis sučelja (IDL) sustava

```
message Result {  
    required string url = 1;  
    optional string title = 2;  
    repeated string snippets = 3;
```

9. U okviru standardnog *WebSocket* programskog sučelja (engl. *WebSocket API*), čitanje poruke na prijemnoj strani izvodi se
- registracijom ugrađene rutine Web preglednika na događaj `onmessage` koja se poziva mehanizmom povratnog poziva
 - registracijom ugrađene rutine Web preglednika na događaj `onmessage` koja se poziva u petlji
 - registracijom korisničke funkcije na događaj `onmessage` koja se poziva mehanizmom povratnog poziva
 - registracijom korisničke funkcije na događaj `onmessage` koja se poziva u petlji
 - automatskim pozivanjem ugrađene rutine Web preglednika, nakon čega se podaci čitaju iz varijable `msg`
10. U sustavu s pravilnim periodičkim promjenama gdje je period promjene dugačak (primjerice, svakih nekoliko sati), a količina korisnih podataka značajno veća od zaglavlja HTTP protokola, promjene s poslužitelja na klijente preporučljivo je izvesti
- tehtnikom prozivanja (*polling*)
 - tehtnikom blokirajućeg prozivanja (*long polling*)
 - protokolom *WebSocket*
 - tehtnologijom *Server-Sent Events*
 - izbor dojavne tehnike nema utjecaja na radna svojstva sustava
11. (10 bodova) Zadane su dvije inačice jedne te iste web usluge s informacijama o predmetu na studijskom programu. Informacije o predmetu sastoje se od: 1) općih podataka o predmetu, 2) podataka o nastavnicima na predmetu i 3) podataka o literaturi. Inačice se razlikuju po načinu organizacije ponuđenih informacija u sredstva poslužitelja. U tablicama su zadane web adrese pojedinih sredstava i veličine tih sredstava za dvije inačice web usluge. Web usluzi pristupa ukupno 1000 studenata na sljedeći način:
- 100 studenata treba sve informacije o predmetu
 - 100 studenata treba samo informacije o nastavnicima, ali svima koji sudjeluju u izvođenju predmeta
 - 600 studenata treba informacije o samo jednom nastavniku, onom u čijoj grupi polazi nastavu
 - 200 studenata treba samo informacije o literaturi.
- Veličina zaglavlja svakog HTTP zahtjeva i svakog HTTP odgovora je 500 okteta (zajedno s praznim retkom na kraju zaglavlja).

Inačica A

URL sredstva	Veličina sredstva
/predmet/osnovno	100
/predmet/nastavnici/nastavnik1	1000
/predmet/nastavnici/nastavnik2	1000
/predmet/nastavnici/nastavnik3	1000
/predmet/literatura	500

Inačica B

URL sredstva	Veličina sredstva
/predmet	3600
/predmet/nastavnici	3000
/predmet/literatura	500

12. (10 bodova) U web usluzi za čavrljanje (*chat*) nalaze se dva klijenta (*klijent 1* i *klijent 2*) koji razmjenjuju tekstualne poruke putem posredničkog poslužitelja. Klijent pošiljatelj poruku šalje poslužitelju, a poslužitelj ju nakon toga isporučuje klijentu primatelju. Oba klijenta mogu poprimiti ulogu pošiljatelja i primatelja. Dinamika slanja poruka s klijenata pošiljatelja prikazana je u tablici. Vrijeme potrebno za prijenos poruka mrežom, kao i za obradu na poslužitelju i na klijentima je zanemarivo. Vrijeme teče od trenutka 0 i odmah nakon toga klijenti i poslužitelj su spremni za rad. Poslužitelj za svakog klijenta pamti samo zadnju poruku koju je taj klijent poslao. Odredite vremena isporuke poruka klijentima primateljima ako se za isporuku poruka koristi:

- prozivanje (*polling*), pri čemu *klijent 1* radi s periodom od 5 sekundi, a *klijent 2* s periodom od 10 sekundi
- blokirajuće prozivanje (*long polling*)
- protokol *WebSocket*.

Odgovore upisati u tablicu.

Poruka	Vrijeme slanja	Klijent pošiljatelj	Vrijeme isporuke klijentu primatelju		
			<i>polling</i>	<i>long polling</i>	<i>WebSocket</i>
1	3	1	3	3	3
2	4	2	5	4	4
3	12	1	X	12	12
4	13	2	15	13	13
5	17	1	X	17	17
6	18	1	20	18	18
7	31	1	30	31	31
8	32	2	X	32	32
9	33	2	X	33	33
10	34	2	35	34	34