

1. (4b)

Za programsko inženjerstvo kao disciplinu koja se bavi izgradnjom složenih programskih sustava:

- (2b) Navedite najmanje tri najvažnija izzova programskog inženjerstva.
- (2b) Navedite najmanje tri najznačajnija modela prog. inž.

2. (4b)

Za modeliranje kao jedan od načina rješavanja problema složenosti:

- (2b) Kratko opisite opći inženjerski model i njegova svojstva.
- (2b) Ukratko opisite osobito svojstvo programa obzirom na mogućnost uporabe modeliranja u odnosu na druge inženjerske discipline.

3. (5b)

Za uprav. proj. prog.-potp. (Software Project Management) kao krovnom akt. prog. inž.

- (3b) Navedite i kratko opisite osnovne elemente fokusa programskog inženjerstva.
- (2b) Prikazite odnose između prethodno definiranih elemenata naznakom vrste odnosa.

5. (2b) Na temelju vašeg iskustva s timskim projektom, odgovorite ukratko na ova pitanja o životnom ciklusu razvoja programske potpore (Software Development Life-Cycle SDLC)

a. (1b) Koja faza nam je bila najteža.  
Zašto?

b. (1b) Koja faza je bila najjasnija.  
Zašto?

6. (4b) Za SDLC:

a. (2b) Navedite tipične faze obuhvaćene generičkim modelom.

b. (2b) Kako su definirane faze modela životnog ciklusa; navedite primjer jedne norme.

7. (4b) Za zahtjeve domene primjene (Domain Requirements) na programsku potporu:

a. (2b) Navedite razloge i ciljeve postavljanja svih zahtjeva i primjer takvog zahtjeva.

b. (2b) Navedite dijeliće oblikovanje programske potpore ključne za izlučivanje svih zahtjeva.

7. (5b) Za inkrementalni pristup razvoju programiske potpore:

- (2b) Opisite svojstva rukovanja zahtjevima.
- (3b) Navedite prednosti inkrementalnog razvoja. Uspoređujući iterativni i uodopadni model obrazložite u kojoj fazi iterativni pokazuje prednost

8. (4b) Za objektno orijentirano modeliranje:

- (2b) Navedite svojstva objektno usmjerene analize i oblikovanja (OOAD)
- (2b) Kratko opisite namjenu i cilj objektno usmjerene analize (OOA)

9. (6b) Za Domain Testing:

- (3b) Odredite minimalan broj ispitnih slučajeva tehnikom ekvivalentnih particija (equivalence partitioning)

```
public String POREKENIrazred(double bruto, double oaksica)
{
    char razred = "*";
    double osnovica = bruto - oaksica;
    if (osnovica < 10000.0)
        razred = "O";
    else if (osnovica < 30000.0)
        razred = "2";
    else
        razred = "E";
    return razred;
}
```

b. (3b) Tko je u programskom timu zadužen za ovaj tip ispitivanja?

Napišite skicu kada je ispitnog slučaja za ovaj pr. uporabom radnog okvira za jedinično ispitivanje (unit testing framework)

10. (4b) Za princip dobrog oblikovanja prog. potp.:

a. (2b) Koje strategije bi koristili kako bi se vedili principom "Povećaj to nečiju"?

b. (2b) Pojasnite princip otvoren - zatvoren (open - closed) kao jednog od pet principa SOLID

11. (4b) Za formalan logički sustav  $\{\Gamma, L\}$ :

a. (2b) Def. pojmove  $\Gamma$  i  $L$ .

b. (2b) Navedite barem četiri primjera pravila generiranja novih formula (pravila prirodnog zaključivanja).

12. (4b) Za radne okvire za web:

a. (2b) Objasnite što je to inverzija upravljanja.

b. (2b) Navedite barem tri temeljne podržane funkcionalnosti.

## Problemski dio - robotska kosilica

13. (5b) UML dijagram obrazaca uporabe:

a. (1b) Navedite korake oblikovanja dijagrama obrazaca uporabe

b. (4b) Modelirajte popratnu mobilnu aplikaciju za nadzor robotske kosilice.

14. (5b) UML - dijagramom razreda modelirajte programsku podršku na kosilici;

15. (5b) UML - dijagramom stanje modelirajte stanja kosilice

16. (5b) Izradite specifikacijski UML - dijagram razmještaja