

# Programiranje 3: Januar 2021

Prezime Ime: \_\_\_\_\_ Broj indeksa: \_\_\_\_\_

## Uputstva

- Ispit traje **90 min**
- Projekat je potrebno kreirati na desktopu i dati mu naziv u formatu: **Prezime\_Ime\_Indeks** (izbegavati naša slova u imenu i prezimenu)
- Dozvoljeno je korišćenje samo onih materijala koji se nalaze u folderu **P3 Jan 2021** na desktop-u
- **Nije dozvoljeno** korišćenje Interneta, niti korišćenje pomoćnih materijala u elektronskom, papirnom, ili bilo kom drugom obliku
- Ovaj list sa tekstom zadatka mora da bude potpisan i ostavljen na tastaturi na kraju ispita
- **Strogo je zabranjeno je iznošenje, fotografisanje ili umnožavanje zadataka** na bilo koji način; povreda ovog pravila se strogo kažnjava
- Preporučuje se često snimanje (Save) onoga što je urađeno, kako u slučaju problema sa računarom, nestankom struje i sl. ne bi bilo izgubljeno ono što je urađeno
- Na kraju rada treba proveriti još jednom da li su sačuvani sve file-ovi u projektu (delovi projekta koji nisu snimljeni neće biti preneti prilikom kopiranja zadataka za pregledanje i naknadne intervencije nisu moguće)
- Zadaci sa kompajlerskim greškama se ne pregledaju i automatski se ocenjuju sa 0 poena
- Nakon završenog ispita treba ostaviti otvoreno razvojno okruženje (dakle, **ne zatvarati ga**)
- Na pitanja tipa "Kada će, otprilike, biti objavljeni rezultati?" u ovom trenutku nije moguće odgovoriti – čim pregledanje bude završeno, rezultati će biti objavljeni na sajtu predmeta i tom prilikom će biti poznat i termin za uvid u radove.

## Zadatak 1

Dat je "todos.json" fajl koji sadrži listu fiktivnih zadataka koji su dodeljeni članovima jednog radnog tima. Svaki zadatak u listi ima sledeću strukturu:

- `userId` - jedinstveni identifikator člana tima kome je zadatak dodeljen,
- `id` - jedinstveni identifikator samog zadatka,
- `title` - opis (fiktivnog) zadatka
- `completed` - boolean indikator da li je zadatak završen ili ne

Koristeći ovaj fajl, odnosno, podatke koje sadrži potrebno je uraditi sledeće:

- Napisati funkciju koja pronalazi i vraća (kao svoju povratnu vrednost) članove tima sa najvećim brojem završenih zadataka. Imati u vidu da može biti više članova tima sa (istim) najvećim brojem završenih zadataka. **(10 poena)**
- Napisati funkciju koja zadacima dodeljuje prioritet kao nasumično generisan ceo broj u opsegu od 1 do 5 (gde 1 ukazuje na najviši prioritet). Konkretno, funkcija bi trebalo svakom rečniku kojim je zadatak predstavljen da doda ključ "priority" a kao vrednost tog ključa postavi slučajno generisani broj između 1 i 5. Tako dopunjen skup podataka upisati u json fajl "prioritised\_todos.json". **(5 poena)**
- Napisati funkciju koja za svakog člana tima pronalazi sve zadatke koje član nije završio i upisuje te zadatke u csv fajl u redosledu prioriteta zadatka (od onih najprioritetnijih do onih sa najmanjim prioritetom) - podrazumeva se korišćenje proritizovane liste zadataka koju kreira prethodna funkcija. Struktura csv fajla bi trebalo da bude sledeća: "task\_id", "task\_priority", "task\_title", dok bi naziv fajla trebalo zadati na osnovu jedinstvenog identifikatora člana tima

## Programiranje 3: Januar 2021

Prezime Ime: \_\_\_\_\_ Broj indeksa: \_\_\_\_\_

(userId). Tako kreirani fajlovi, za svakog člana tima (koji nije završio svoje zadatke), bi trebalo da budu sačuvati u istom direktorijumu (npr. "member\_assignments"). **(15 poena)**

Proveriti sve kreirane funkcije u main delu skripte (tj u okviru 'if \_\_name\_\_ == "\_\_main\_\_":'). Sve provere treba da prikazuju i odgovarajuće tekstualne prompt-ove ili prateće poruke iz kojih se jasno vidi na šta se proverava odnosi. **(15 poena)**

### Zadatak 2

Kreirati dekorator koji 'loguje' podatke o radu funkcije u csv fajlu "execution\_log.csv" tako što upisuje sledeće podatke odvojene tačka zarezom:

- trenutak (datum i vreme) poziva funkcije
- pun naziv funkcije (uključujući i njene ulazne argumente)
- vreme izvršavanja funkcije (koju dekoriše) u milisekundama

Potrebno je obezbediti da se sadržaj loga ne prepisuje pri svakom novom pozivu dekorisane funkcije već da se postojeći sadržaj log fajla samo dopunjuje.

Imati u vidu da dekorisana funkcija može imati nula ili više pozicionih argumenata (args), kao i nula ili više imenovanih argumenata (kwargs). **(20 poena)**

Testirati dekorator tako što ćete funkcije iz prethodnog zadatka dekorisati ovim dekoratorom. **(5 poena)**

Predlog je da se zadatak rešava korišćenjem onih pristupa i Python paketa koji su korišćeni u sličnim zadacima u toku nastave, ali je dozvoljeno rešiti ga i na drugi način.