

Drugi kolokvijum - teorijski deo-

Napomena: Upotreba štampane literature i komunikacionih uređaja je zabranjena i strogo kažnjiva. Dozvoljena je samo upotreba elektronskih materijala sa diska *Materijali*. Kolokvijum traje 135 minuta.

Broj indeksa	Ime i prezime	Broj poena	
		1	2

1. [3] Objasniti zašto sledeći kod narušava D u SOLID principima. Refaktorizovati dati kod da poštuje SOLID principe (koristiti pri tome ABC modul u pythonu da bi se jasno naznačilo kakve vrste je koja klasa). Pri tome dodatno uvesti i mogućnost da `ElectricPowerSwitch` kontroliše i rad još jednog uređaja (ventilatora).

```
class LightBulb:
    def turn_on(self):
        return "Light bulb turns on"

    def turn_off(self):
        return "Light bulb turns off"

class ElectricPowerSwitch:
    def __init__(self, l: LightBulb):
        self.light_bulb = l
        self.on = False

    def press(self):
        if self.on:
            self.on = False
            return self.light_bulb.turn_off()
        else:
            self.on = True
            return self.light_bulb.turn_on()
```

2. [2] Uvesti kontejner sa inverzijom kontrole (IOC) za klase od interesa u prethodnim primeru.

Drugi kolokvijum - praktični deo -

Napomena: Upotreba štampane literature i komunikacionih uređaja je zabranjena i strogo kažnjiva. Dozvoljena je samo upotreba elektronskih materijala sa diska *Materijali*. Ispit traje 165 minuta.

Broj indeksa	Ime i prezime	Broj poena

3. [10] Uvesti kontejner sa inverzijom kontrole (IOC) za klase od interesa u prethodnim primer. Potrebno je napraviti veb aplikaciju koristeći arhitekturni projektni uzorak *MVC (Model-View-Controller)* u programskom jeziku *Python*, bez korišćenja bilo kog radnog okvira (*framework*). Dozvoljeno (i poželjno) je korišćenje biblioteka, korišćenih u primerima sa vežbi, a u nastavku su izdvojene glavne biblioteke i moduli:

- biblioteka `wsgiref` za implementaciju serverske aplikacije,
- modul `cookies` iz biblioteke `http` za implementaciju kolačića i
- modul `mysql.connector` za konekciju sa bazom podataka.

Aplikacija treba da sadrži tri veb stranice:

- jednu za prijavljivanje korisnika u sistem (`login.html`),
- drugu za prikaz recepata (`recepti.html`) i
- treću za dodavanje recepta (`dodaj_recept.html`)

Podrazumevano se učitava stranica za prijavljivanje korisnika, osim ako korisnik nije već prijavljen. Kada se prijavi, korisnik prelazi na stranicu za prikaz svih recepata. U nastavku se nalazi spisak stavki koji je potrebno realizovati (a pored svake stavke je naznačeno koliko poena nosi):

- [1] Podignut je server, koji otvara podrazumevano `login.html` stranicu. Takođe je potrebno vratiti odgovarajući status u odgovoru.
- [1] U slučaju da se zahteva nepostojeća putanja od aplikacije, potrebno je vratiti informaciju u vidu HTML stranice o tome, sa odgovarajućim statusom.
- [1] Logika za prijavljivanje korisnika je implementirana, tako da se ispisuju poruke korisniku u slučaju da ne postoji nalog sa korisničkim imenom, ili ukoliko lozinka nije dobra.
- [1] U slučaju uspešnog prijavljivanja korisnika u sistem, korisnik se prebacuje na stranicu `recepti.html` u sesiji.
- [1] Sesija će biti zapamćena i ispravno će raditi u slučaju da se server (aplikacija) ponovo pokrene (resetuje).
- [1] Recepti se izlistavaju ispravno iz baze, po datom HTML šablonu (`recept-sablon.html`).
- [1] Recepti se ažuriraju asinhrono u slučaju da je dodat novi.
- [1] Na stranici za dodavanje recepta korisnik može da dodaje jedan po jedan sastojak, i tekući recept sa svim do tada dodatim sastojcima se pamti u kolačićima i ispisuje u definisanoj sekciji stranice, sve dok korisnik ne upiše naziv recepta i klikne dugme Završi recept.
- [1] Kada korisnik klikne na dugme Završi recept, recept sa svim dodatim sastojcima se uspešno upisuje u bazu i korisnik se prebacuje na `recepti.html`.
- [1] U slučaju pokušaja prelaska na stranicu za prijavljivanje, a da je korisnik već prijavljen u sistemu, radi se automatsko prebacivanje na `recepti.html`.

Napomene:

- U slučaju da aplikacija nije implementirana u skladu sa *MVC* projektnim uzorkom, dobija se maksimalno 50% osvojenih poena.
- U slučaju da aplikacija nije implementirana korišćenjem baze podataka, dobija se maksimalno 50% osvojenih poena.
- Pošto vrednosti kolačića mogu biti samo stringovi, na raspolaganju su dve funkcije iz json modula za serijalizaciju Python rečnika:
 - `json.dumps(dict)`: str, koja serijalizuje Python rečnik poslat kao argument u string
 - `json.loads(str)`: dict, koja string koji predstavlja serijalizovan Python rečnik deserijalizuje
- Na disku *Matrijali* (N:) možete naći početni sadržaj projekta sa mysql bibliotekom u okviru ZIP arhive `project_with_mysql_connector.zip`
- Predaja se vrši kopiranjem foldera sa projektom na disk *Rad* (L:).