

## **Verifikacija i validacija softvera**

### **Projektni zadatak – I dio**

**Ukupan broj bodova: 10**

Datum objave: 28.10.2025.

Sav napisani kod i popratne materijale staviti na repozitorij kreirane *Azure DevOps* organizacije projekta na koji je potrebno poslati pozivnicu asistentici. Kreiranu organizaciju imenovati u formatu Ime1\_Ime2\_Ime3\_Ime4.

#### **Task 1: Specifikacija i razvoj programskog rješenja**

Rok za specifikaciju: 2. laboratorijska vježba

Rok za implementaciju: 3. laboratorijska vježba

Za potrebe projekta potrebno je napisati kratku specifikaciju za dodijeljenu temu i razviti srednje složeno konzolno programsko rješenje u .NET okruženju.

Programsko rješenje treba zadovoljavati sljedeće:

1. Funkcionalnosti/modelskih klasa/servisa treba biti onoliko koliko ima članova tima,
2. Svaka od funkcionalnosti treba imati barem jednu metodu sa vlastitom implementacijom složenog algoritma (sortiranje, pretraživanje, izračunavanje, kriptografija, i sl.),
3. Pravilno upravljanje izuzecima,
4. Dobro dizajnirano rješenje (korištenje interfejsa, nasljeđivanja, itd.),
5. I/O kroz konzolu – nema potrebe za korisničkim interfejsom,
6. Sama implementacija se treba odvijati u skladu sa preporučenim načinom rada (koristeći Git branches, i pull requests).

Napomene:

1. Perzistencija podataka nije potrebna,
2. Svaki tim će implementirati svoje programsko rješenje po temi dodijeljenoj tom terminu vježbi,
3. Obavezna predaja specifikacije sa dijagramom klasa na laboratorijskoj vježbi 2. Na istim će se provesti diskusija sa asistenticom o eventualnim promjenama prije implementacije,
4. Predaja i prezentacija implementacije se izvodi na 3. laboratorijskoj vježbi.

Raspored tema:

Grupa 1	Aplikacija za tendere
Grupa 2	Aplikacija za vođenje inventara
Grupa 3	Aplikacija za preporuku filmova ili knjiga
Grupa 4	Aplikacija za raspored učenja i ispitnih rokova
Grupa 5	Team Task Manager – praćenje zajedničkih zadataka u timu
Grupa 6	Aplikacija za planiranje putovanja

### **Task 2: Inspekcija koda (2 boda)**

U sklopu laboratorijske vježbe 4 izvršiti inspekciju koda drugih timova u grupi.

Programski kod aplikacije nad kojom će se izvršavati inspekcija postaviti u novi repozitorij u svojoj *Azure DevOps* organizaciji, i kreirati *pull request* na način kako je objašnjeno na laboratorijskim vježbama. Potrebno je ilustrirati dodavanje komentara i kreiranje *issue* kartica.

Unutar svakog tima odrediti moderatora inspekcije. Moderator treba da dodijeli kodox pojedinim članovima tima za inspekciju i odgovarajuće liste provjere (*check liste*). U svakom komentaru i *issue* kartici je potrebno naznačiti visinu ozbiljnosti greške, te stavku unutar odgovarajuće liste provjere u sklopu koje je pronađena greška.

Nakon provedene inspekcije, napisati kratki izvještaj o inspekciji i uraditi Pareto analizu uzroka i defekata u pronadjenim greškama. Izvještaj postaviti u isti repozitorij kao i programski kod aplikacije.

### **Task 3: Unit testiranje (6 bodova)**

Rok za Task 3: 5. laboratorijska vježba

**3.1** Svaki član tima treba da napiše *unit* testove za dijelove koda koje je napisao (implementirao) i postigne maksimalan obuhvat koda (minimalno 90%). Klasu Program nije potrebno testirati. Testovi treba da se odnose na testiranje osnovnih funkcionalnosti koda i izuzetaka. Obavezno primijeniti i *data driven* testiranje sa pohranjenim podacima u najmanje 2 forme. Testovi treba da budu dobro dizajnirani i da poštuju sve principe navedene na predavanjima i vježbama (dobro imenovani, dobro organizirani, bez programske logike, međusobno neovisni,...). Iskoristiti što više stereotipa za testne metode koji su navedeni u predavanju.

**3.2** Potrebno je da iskoristiti zamjenske objekte za module koji nisu još implementirani i ilustrirati *unit* testiranje sa zamjenskim objektima. Svaki član tima treba da napravi barem 1 test koji koristi zamjenski objekat.

Izvještaj treba sadržavati spisak testova koje je kreirao svaki član tima, opis funkcionalnosti koja se testirala koristeći *data driven* testiranje i zamjenske objekte, i prikaz pokrivenosti koda za svaku od klase.

### **Task 4: TDD razvoj (2 boda)**

Rok za task 4: 5. laboratorijska vježba

Specificirati 2 nove funkcionalnosti za aplikaciju koje je potrebno implementirati primjenom TDD pristupa. Organizovati tim tako da svi članovi učestvuju u pisanju testova i implementacije.

Izvještaj treba sadržavati specifikaciju novih funkcionalnosti i način njihovog razvijanja u odnosu na napisane testove u sklopu TDD razvoja.