Laboratorijska vježba br. 1

**Izvještaj o Inspekciji Koda**

***Check liste inspekcije koda***

*Označiti stavke na listama za inspekciju koje su ispunjene, nakon vršenja inspekcije koda. Za stavke koje se ne označe potrebno je navesti detaljne informacije o greškama u nastavku.*

**Inspekcija strukture programskog rješenja**

*Ova lista stavlja fokus na potencijalne probleme u strukturi programskog koda, analizi koda na visokom nivou i poštovanju standarda.*

Kod je napisan u skladu sa važećim standardima kodiranja.

Stil kodiranja je konzistentan u cijelom programskom rješenju.

Kod je ispravno formatiran.

U kodu nema funkcija koje se ne pozivaju ni na jednom mjestu.

Nema nedostižnih linija koda.

Nema bespotrebnog implementiranja funkcija koje mogu biti zamijenjene postojećim bibliotekama.

U kodu nema ponavljanja koje može biti zamijenjeno jedinstvenom funkcijom.

* Moglo bi biti prilika za refaktoriranje i izdvajanje zajedničkog koda u odvojene funkcije, ali to ovisi o složenosti programa i ciljevima projekta.

Memorija se koristi na efikasan način.

Nema korištenja *magičnih brojeva* i konstanti bez korištenja varijabli.

* Magični brojevi se koriste u kodu, kao što su brojevi 200M, 150M, 0.02M, itd. Preporučljivo je izdvojiti ove konstante kao varijable s opisnim imenima radi bolje čitljivosti i održavanja.

Nema previše dugih i kompleksnih blokova koda.

* Kod može biti nešto kompleksan, posebno u izračunima cijene najma. To ovisi o funkcionalnostima koje su potrebne u projektu.

**Inspekcija dizajna programskog rješenja**

*Ova lista stavlja fokus na potencijalne probleme u poštovanju objektno-orijentisanih principa, SOLID principa i dizajn pattern-a u okviru programskog rješenja.*

Svaka klasa ima malu kompleksnost i jedan tip operacija i zaduženja.

* Klase Stan, NenamjestenStan i NamjestenStan čine se relativno jednostavnima i imaju jasne zadatke

Klase su prilagodljive budućim promjenama.

Svi objekti izvedenih klasa zamjenjivi su svojim osnovnim klasama.

* To bi moglo biti problematično u ovom slučaju jer izvedene klase NenamjestenStan i NamjestenStan imaju dodatna svojstva (VrijednostNamjestaja i BrojAparata) koja osnovna klasa Stan nema.

Interfejsi su jednostavni, s malim brojem funkcija.

Dubina nasljeđivanja nije velika.

* Dubina nasljeđivanja u ovom kodu nije velika, jer se koristi samo jedna razina nasljeđivanja (NenamjestenStan i NamjestenStan nasljeđuju Stan). Međutim, ovisno o potrebama projekta, može se razmotriti korištenje više razina nasljeđivanja.

Klijent može jednostavno pristupati objektima kontejnerskih klasa, bez potrebe definisanja detalja gradivnih dijelova klase.

U slučaju potrebe ponovnog korištenja većeg broja istih objekata, objekti se ne instanciraju više puta.

* U slučaju potrebe ponovnog korištenja većeg broja istih objekata, objekti se ne instanciraju više puta.

Instanciranje kontejnerske klase vrši se samo jednom.

Sigurnost aplikacije osigurana je putem *proxy*-a.

* Ovaj aspekt se ne primjenjuje na trenutni kod jer se ne koristi koncept proxyja

**Inspekcija varijabli i izraza programskog rješenja**

*Ova lista stavlja fokus na potencijalne probleme u strukturi koda na visokom nivou, uključujući varijable i izraze u kodu.*

Sve varijable imaju imena koja odgovaraju njihovoj namjeni.

Koristi se jedan stil imenovanja varijabli.

Nema varijabli koje se ne koriste.

Nema neosiguranih potencijalnih dijeljenja s nulom.

* Kod ne sadrži izričite provjere na null vrijednost, što može uzrokovati neželjeno ponašanje u slučaju da se varijable koriste prije nego što im se dodijeli vrijednost

Operator **=** ne koristi se u logičkim izrazima.

* Operator = se koristi u izrazima za dodjelu vrijednosti varijablama, što je ispravno, ali kod može sadržavati logičke izraze koji sadrže operator =, što može biti potencijalno problematično

**Inspekcija petlji i grananja programskog rješenja**

*Ova lista stavlja fokus na potencijalne probleme u petljama i grananjima u kodu.*

Nema praznih niti nedostižnih blokova koda.

U *if* blokovima testiraju se češći scenariji.

Svi *switch* iskazi imaju definisan *default* slučaj.

* U kodu nema switch iskaza pa ova tačka nije primjenjiva na trenutni kod

Sve petlje imaju uslov završetka.

Nema velikog broja gniježdenja petlji.

U petljama nema koda koji se može izvršiti izvan petlje.

**Inspekcija memorijskih operacija programskog rješenja**

*Ova lista stavlja fokus na potencijalne probleme u korištenju memorije te konekciji s bazama podataka, vanjskim uređajima i korištenjem file-ova u kodu.*

Sve varijable koje koriste indeksiranje su inicijalizirane prije korištenja.

Sva alocirana memorija dealocira se prije završetka izvršavanja.

Pri radu s vanjskim uređajima, postoji provjera za *timeou0t*.

* Kod se ne čini da se bavi radom s vanjskim uređajima koji zahtijevaju timeout provjeru

Prije pokušaja modificiranja *file*-ova, provjerava se da li oni postoje.

* U kodu nema operacija za modificiranje ili provjeru postojanja datoteka, pa ova tačka nije primjenjiva na trenutni kod

Nakon završetka transakcije, konekcija s bazom podataka se uvijek zatvara.

* Trenutni kod ne uključuje rad s bazom podataka, pa ova tačka nije primjenjiva na trenutni kod

**Inspekcija dokumentacije programskog rješenja**

*Ova lista stavlja fokus na potencijalne probleme u razumljivosti i jednostavnosti dokumentovanja koda.*

Svi kompleksni dijelovi koda posjeduju komentare.

* U trenutnom kodu nije vidljivo da su svi kompleksni dijelovi koda opisani komentarima

Dijelovi koda podijeljeni su u regije.

* Trenutno kod ne koristi regije za organizaciju dijelova koda. Korištenje regija može pomoći u preglednosti i organizaciji koda

Metode klasa imaju svoje opise.

U cijelom rješenju koristi se jedan stil komentarisanja koda.

***Informacije o timu koji vrši inspekciju koda***

*Popuniti informacije o članovima tima koji vrši inspekciju.*

***Informacije o timu koji vrši inspekciju koda***

*Popuniti informacije o članovima tima koji vrši inspekciju.*

Ime i prezime, broj indexa: Eldar Muratović, 145-ST

Zaduženje: Inspekcija Program i Stan klase

Predmet inspekcije: Checklista

Ime i prezime, broj indexa: Vedad Ribo, 154-ST

Zaduženje: Inspekcija Nenamjesten Stan klase

Predmet inspekcije: Checklista

Ime i prezime, broj indexa: Hamza Bunar, 165-ST

Zaduženje: Inspekcija Namjesten Stan

Predmet inspekcije: Checklista

***Izvještaj o metrikama grešaka***

Ukupan broj pronađenih grešaka: Click here to enter text.

Normirani broj grešaka: Click here to enter text.

Broj grešaka po LOC: Click here to enter text.

Broj normiranih grešaka po LOC: Click here to enter text.

Efikasnost otkrivanja grešaka: Click here to enter text.

Normirana efikasnost otkrivanja grešaka: Click here to enter text.