UNIVERZITET U SARAJEVU
ELEKTROTEHNIČKI FAKULTET SARAJEVO

DOMAĆA ZADAĆA br. 1

-RAZVOJ PROGRAMSKIH RJEŠENJA-

Prezime i ime: ŠEHALIĆ MIRZA
Broj indeksa: 17324
Grupa: 3
Odsjek: RI

Potpis studenta: _____

1 Dokumentacija

Dobro došli u dokumentaciju za projekat Zadaće 1 iz predmeta Razvoj programskih rješenja, autora Mirze Šehalića.

1. Što se tiče implementacije prve stavke, implementiran je model klasa koji se efektivno sastoji od seta klasa koje su potrebne za pacijente klinike (tri *partial* klase Pacijenta), seta klasa koje predstavljaju uposlenike klinike (Uposlenik, Doktor i Tehnicar), seta klasa koje su potrebne za realizaciju ordinacija klinike (Ordinacije sa interfejsom) i klase koja odgovara pregledima (Pregled.cs). Nasljedjivanje je obavljeno gdje god je ono bilo moguće, što se specifično odnosi na nasljedjivanje uposlenika i pacijenata po vrstama. Što se tiče specifičnih tipova klasa, klasa Pregled.cs je implementirana kao *sealed class*, dok je za klase Ordinacija, OrdinacijaDermatolog, OrdinacijaOrtoped te OrdinacijaKardiolog implementiran i interfejs i nasljedjivanje iz bazne klase Ordinacija (za posljednje tri), kako bi se izvukao maksimum iz situacije i stvorila kvalitetna hijerarhija koja omogućava i nadopunjavanje sa dodatnim klasama. Već smo spomenuli da je klasa za Pacijenta podijeljena na tri *partial* klase zbog obima metoda, konstruktora i anamneza predvidjenih za odredjenog pacijenta.

2. Traženi interfejs je implementiran kao vodilja za ordinacije, obzirom da je u zadaći naglašena potreba za dodavanjem ordinacija u nekoj bližoj budućnosti razvoja klinike, a znamo da to implicira ili apstraktnu klasu ili interfejs, ovaj put sam izabrao potonje.

3. Prilikom implementacije pretrage kartona po raznim kriterijima iskoristio sam delegate (te i lambda izraze) koji su zaista pojednostavili pomalo konfuzne upite u formi lambda izraza. Lambda izrazi su korišteni nekoliko desetina puta u Program.cs, doslovno gdje god je to općenito bilo moguće radi kraćeg zapisa i efektivnijeg izvršavanja koda.

4. Za spašavanje podataka su korištene kolekcije i to striktno generičke, obzirom da je poznato da boxing i unboxing iz običnih kolekcija nepotrebno usporava program. Takodjer za upravljanje listama čekanja pacijenata (opcija 2 glavnog menija) i sortiranje istih je korišten Queue<T> koji posjeduje vrlo zgodne metode Enqueue i Dequeue, čiji FIFO princip je upravo bio potreban za odgovarajući zadatak. Za sistem naplate je korištena lista tupleova koja pamti postupak i uzima cijenu postupka iz odgovarajuće ordinacije kojoj pripada doktor koji je izvršio pregled.

5. U sklopu odvojenog .dll fajla je kreirana funkcionalnost sa uposlenicima, preciznije doktorima i tehničarima. Bazna klasa je Uposlenik.cs, dok se iz nje izvode Doktor.cs i Tehnicar.cs kao podvarijante uposlenika. Obzirom da postoji odredjena hijerarhija klasa, ne bi trebalo biti teško dodati i upravu ili neke druge tipove uposlenika, ako se ukaže potreba.

6. Izvršena je dekompozicija klasa, interfejs je izdvojen iz Program.cs u zaseban fajl ProgramMain.cs, koji je zbog toga deklarisan kao *partial class*. Svaka klasa je u zasebnom fajlu te su poštovani osnovni principi iz predavanja i postavki LV.

7. Kao što smo prethodno naveli, glavni meni je u zasebnom fajlu u odnosu na metode iz Program.cs. Programski meni je realiziran kao u specifikaciji, pri čemu se odgovarajuće opcije odnose na:

7.1. Prva opcija se odnosi na registraciju pacijenta gdje se unose podaci iz postavke te je implementirana provjera validnosti matičnog broja (na prvih 7 cifara koje odgovaraju datumu rodjenja, na pozitivnost i ispravan cijeli broj, te na ukupno 13 cifara). Nakon unosa osnovnih podataka pacijentu se unose pregledi koje želi obaviti, a sistem ih automatski nudi iz raspoloživih ordinacija. Takoer ukoliko se radi o hitnom pacijentu, doktor može upisati sve potrebne podatke o hitnom pregledu te statusu pacijenta i tome da li je on preživio pregled, kao što je definirano u postavci. Funkcija je case-insensitive kao i svaka buduća koja radi sa pacijentima, tako da nema problema oko toga da li je doktor unio malo početno slovo imena/prezimena.
7.2. Druga opcija se odnosi na kreiranje rasporeda pregleda za pacijente što se, kao što smo ranije opisali, vrši putem generičkog reda (Queue<T>). Funkcija se oslanja na korištenje lambda izraza i LINQ ekspresija, te na vrlo efikasnu metodu sortiranja koja prima delegat na Tuple sa tri člana i sortira red svakim pozivom funkcije, tako da ukoliko neki prezauzeti doktor vrši svoje preglede brže, njegova ordinacija odredjenom pacijentu može izaći kao iduća za posjetu, čak i prije nekog doktora gdje je pacijent drugi ili treći u redu čekanja, ali pregledi teku sporo. Takodjer se pazi da se ne ispisuju ordinacije za koje se pacijent nije prijavio.
7.3. Treća opcija se odnosi na kreiranje kartona za pacijente. Glavni preduslov za kreiranje jeste da je pacijent već evidentiran u odgovarajućoj listi. Ukoliko to nije tako, korisniku aplikacije će se ponuditi opcija da kreira novog pacijenta i onda nastavi gdje je stao sa kreiranjem kartona. Takodjer prije samog kreiranja kartona se vrši i obavezna anamneza, kao u specifikaciji.

7.4. Četvrta opcija se odnosi na pretragu kartona gdje je implementiran delegat, kao što je prethodno i objašnjeno. Funkcija će prvo provjeriti da li postoji pacijent i njegov karton, a ukoliko se ustanovi da ne postoji, ponudit će kreiranje nedostajućih elemenata i nastaviti rad tamo gdje je stala.
7.5. Peta opcija se odnosi na registraciju pregleda, gdje se takodjer pazi da li je kreiran pacijent i njegov odgovarajući karton i postupa kao u prethodnoj stavci ukoliko to nije istina. Doktori unose potrebne podatke za kreiranje pregleda oko samih detalja pregleda, nakon čega unesene podatke moraju verificirati validnom licencom. Ukoliko licenca nije validna, unos se ponavlja dok se ne unese validna licenca ili doktor odustane od pregleda.
7.6. Šesta opcija se odnosi na analizu poslovanja klinike kroz tri stavke, kao u specifikaciji. Odabrane stavke za ovaj program su: prosječan broj pregledanih pacijenata po doktoru, prosječan iznos računa u klinici te udio pušača i alkoholičara medju pacijentima.
7.7. Sedma opcija se odnosi na stavku naplate pregleda pacijentima, prilikom čega svaka ordinacija ima svoju cijenu pregleda koja ovisi o tome da li je u pitanju rutinski pregled kao pregled za vozački ispit ili sistematski pregled u okviru firme ili specifični pregled koji je pacijent zakazao. Implementirani su svi popusti i poskupljenja u skladu sa postavkom.

U ČITAVOM PROGRAMU SU REALIZIRANE VALIDACIJE PODATAKA, KAKO PROGRAM NE BI PADAO RADI EVENTUALNOG POGREŠNOG UNOSA!