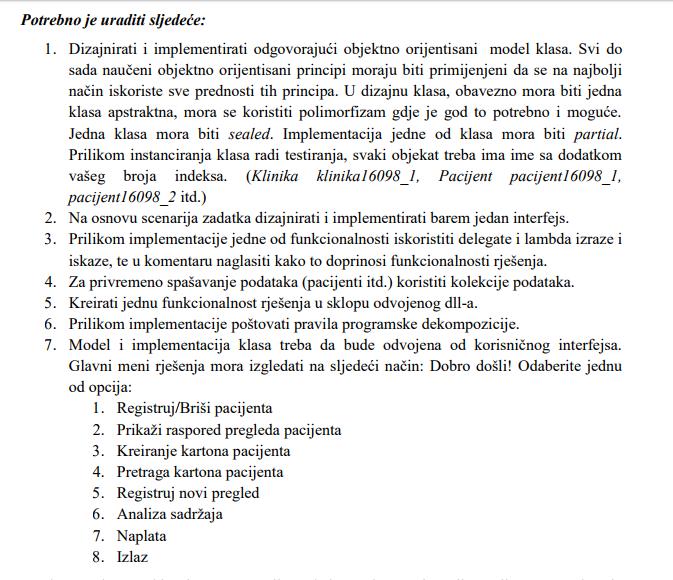
Isječak iz postavke zadaće:



1. Što se tiče prve stavke zahtjeva, ispoštovan je zahtjev da se svaka klasa prilikom instanciranja imenuje sa tipom i odgovarajućim brojem indeksa (17818 u mom slučaju). Implementirano je i odgovarajuće nasljeđivanje gdje god je ono bilo smisleno (npr. naslijedila sam **HitniSlucaj.cs** iz **NormalniPacijent.cs,** obzirom da je HitniSlucaj podvrsta tipa pacijenta). Također je u samom **Program.**cs korišten polimorfizam kod svake instance i funkcijskog poziva pacijenta. Implementirana je i apstraktna klasa za evidenciju tipova uposlenika (**Uposlenik.cs**), kako bi se u budućnosti olakšalo dodavanje novih tipova uposlenika (ukoliko to bude potrebno). Također je spriječeno i instanciranje tipa Uposlenik, jer smatram da isto nije potrebno u ovom rješenju. Obzirom na obim klase Doktor i činjenicu da nije potrebno nasljeđivati i praviti podvrste doktora, odlučila sam se da ona bude tipa **sealed partial class**, pa je tako dosta jasnije shvatiti samu pozadinsku logiku iza navedenih klasa.
2. Ispoštovan je zahtjev za postojanjem barem jednog interfejsa (**InterfejsOrdinacije.cs**), smatram da je to najpogodnije bilo učiniti kod svih ordinacija te je sada dosta pojednostavljeno dodavanje planiranih ordinacija za tehničare.
3. Delegati su iskorišteni kod registracije pacijenta kada pacijent bira željene preglede u metodi Registruj klase **Program.cs**, rješenje je sada dobilo na održivosti te smatram to dobrim primjerom za upotrebu delegata. Lambda iskazi su korišteni u pojedinim dijelovima samog modela klasa.
4. Za spašavanje podataka koristila sam kontejnersku klasu **Klinika.cs**, u kojoj su redom definirane liste pregleda, doktora i pacijenata, instance doktora kreirane radi testiranja programa te liste svih redova čekanja postojećih klinika. Upotreba kontejnerske klase je olakšala slanje parametara u metode i doprinijela održivosti rješenja.
5. Kreirala sam Class library (.dll) u kojem su smještene ordinacije svih doktora, kao i bazna klasa **Ordinacija.cs** te pripadajući interfejs **InterfejsOrdinacije.cs.**
6. Izvršena je propisna dekompozicija te su sve klase u zasebnim fajlovima, nema overengineeringa i nepotrebnih klasa (barem po mom mišljenju).
7. Implementiran je meni u skladu sa postavkom problema te je isti funkcionalan.

Implementiran je i raspored obilaska ordinacija po prioritetima kao i tri metode anallize poslovanja klinike.