Zadaća 2

**Izvještaj o zadaći**

U zadaći 2 sam implementirao sve zadane zadatke.

Prvi zadatak je bio timski rad, gdje smo pored 6 algoritama dodali još 3 algoritma. Sva 3 rade korektno. Promijenili smo i strukturu cijelog main.cs. Dodali smo i klasu Osobe, sa preklopljenim operatorima i osnovnim funkcionalnostima kao što je dobijanje random osobe, tj. osobe sa random atributima i slično. Prilikom izrade smo primijetili kako određeni algoritmi zahtjevaju već sortirani niz, tako da sam individualno i pored ispisa sa prethodne zadaće dodao kolonu koja prikazuje da li i-ti algoritam sortira tj. modificira kolekciju ili ne.

Drugi zadatak je bila individualna implementacija zamjenskog objekta, kojeg sam realizovao koristeći Fake objekat. Napravio sam interface koji sam kasnije naslijedio i implementirao dodavajući tri funkcije, koje prate kapacitet, broj učitanih vrijednosti, te funkciju koja simulira učitavanje vrijednosti iz neke baze. Napisao sam testove i za taj objekat iako nije bilo potrebno. Pored toga, klasa koja implementira interface, ima dva atributa: 'ucitano’ koji ima getter i setter, dok ‘kapacitet’ ima samo getter, jer setter i nema nekog smisla, s obzirom da sama klasa ima konstruktor preko kojeg se postavlja kapacitet. Funkcija koja simulira čitanje iz baze baca izuzetak ukoliko se pokuša učitati više elemenata nego što je vrijednost kapaciteta.

Što se tiče trećeg zadatka napisao sam 61 test, gdje testiram sve funkcije klasa: Main, Pretrage, Osoba. Koristio sam XML file za Data Driven testove iz zadatka, i unio sam 5 testnih osoba za koje sam vršio Assertove. Koristio sam TestInitialize, kao i TestCleanup. Dostigao sam ukupan code coverage od 60%. Naime, procenat izgleda mal, međutim problem nastaje kada se treba testirati dio koda gdje postoji Console.ReadLine, a u nekim djelovima i Console.WriteLine. Generalno, mnogo sam vremena proveo čitajući zašto je to tako i izgleda da unit test template koji mi koristimo ne dozvoljava otvaranje konzole pri pokretanju testova, što naravno samim time onemogućuje ikakav unos sa tastature. Ispis će raditi za neke male vrijednosti, međutim za veće ispise gdje se koristi formatirani ispis ne radi. Jedan od razloga je što je naš ispis preširok za output unit testova. Samim time, nisam mogao dostići code coverage od 80% međutim situacija code coverage-a po klasama je sljedeća:

* Klasa ‘Testovi’ = 97.7% / 100%
* Klasa ‘Pretrage’ = 94% / 100%
* Klasa ‘Osoba’ = 93.8% / 100%
* Klasa ‘main’ = 25% / 100%

Iako je code coverage klase 'main' poprilično mali, sve funkcije koje imaju neku logiku, i ima smisla testirati su testirane neovisno od funkcija koje vrše ispis ili unos sa tastature.

Funkcije koje imaju formatirani ispis i/ili koriste Console.ReadLine, a time onemogućuju testiranje su:

* ZapisiUFile,
* IzvrsiPretragu,
* IspisiListing,
* ZapisiSveUFile,
* OdradiZaListu,
* OdradiZaNiz,
* IspisiOsobu,
* KorisnickaPretraga,
* TakmicenjeAlgoritama,
* Main

Broj linija svih ovih funkcija zajedno je 420, dok je ukupan broj linija main.cs 562.

420/562 = 0.747330… = 75% što nam uz prijašnju informaciju da imam code coverage od 25% u main.cs govori da sam istestirao sve osim nabrojanih funkcija iz spomenutog razloga.