Projektna dokumentacija- OOP

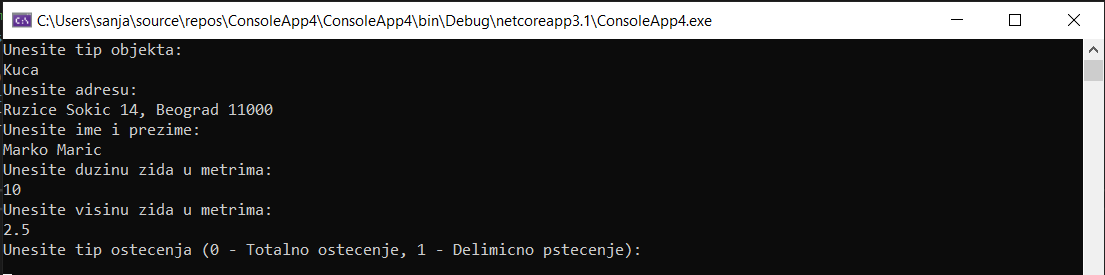
**Uvod-O projektu**

Firma “Poligon” se bavi popravkom fasada. U svrhe posla dizajnirana je aplikacija koja korisniku omogucava racunanje cene popravke stete, u zavisnosti od vrste ostecenja. U koliko mu odgovara cena korisnik salje zahtev firmi za popravku i na taj nacin se uspostavlja interakcija izmedju korisnika I firme.

Aplikacija radi na sledeci nacin:

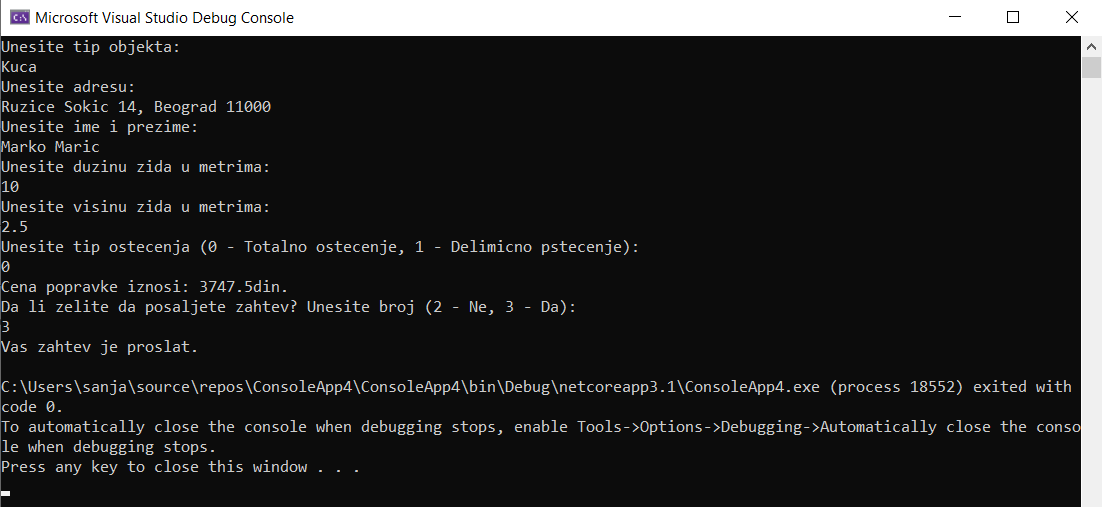
Korisnik prvo unosi tip objekta. Objekat moze biti kuca, zgrada, skola, drzavna ustanova, itd. Zatim unosi adresu, kao i svoje ime i prezime.

Potom unosi podatke o visini I sirini zida u metrima, na kome se nalazi ostecenje. Na osnovu svoje samostalne procene zakljucuje da li je neophodna totalna ili delimicna popravka zida.



Slucaj 0:

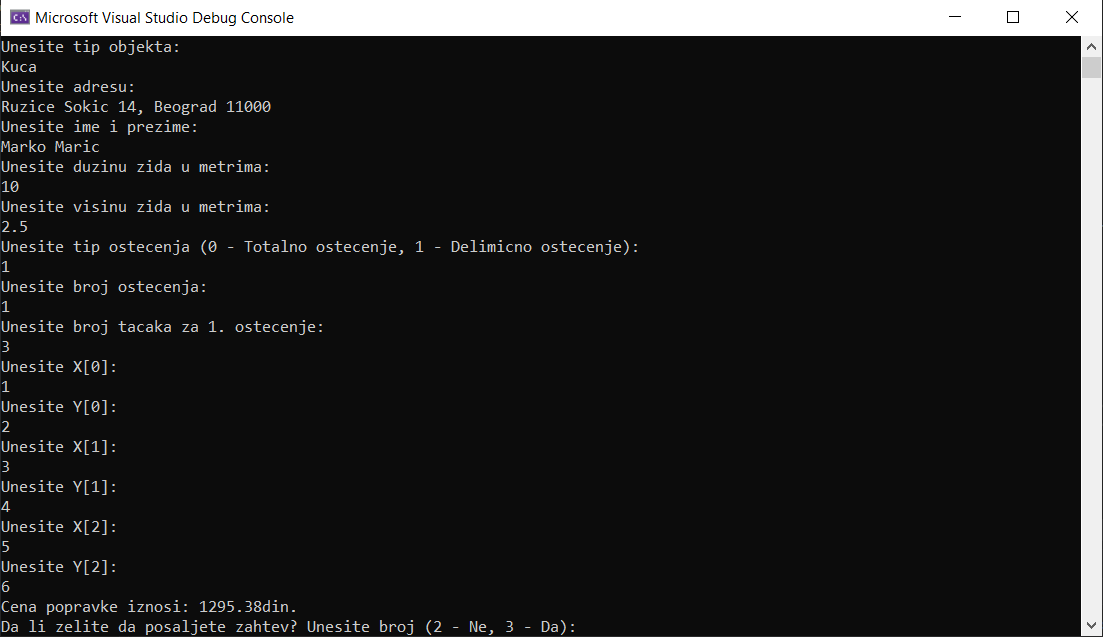
Korisnik je procenio da je potrebna totalna popravka zida. Aplikacija na osnovu unetih podataka racuna cenu, tako sto izracunava povrsinu zida i mnozi sa cenom farbe, potrebne za saniranje stete. Sledeci upit na koji korisnik odgovara je: ”Da li zelite da posaljete zahtev?”. Ukoliko zeli da posalje zahtev, aplikacija tom prilikom prosledjuje zahtev firmi, u suprotinom se prekida izvrsavanje aplikacije.



Slucaj 1:

Korisnik je procenio da je potrebna delimicna popravka zida. Prvo unosi broj ostecenja na zidu (broj ostecenih mesta na zidu). Zatim unosi broj tacaka za svaki od ostecenja. Mora voditi racuna da je minimalan broj unetih tacaka 3, a maksimalan 9. Ukoliko pogresi, aplikacija mu prijavljuje gresku sa objasnjenjem. Posle adekvatnog unosa broja tacaka, aplikacija zahteva unos koordinata kako bi se formirala povrsina poligona. Prilikom popravke pojedinačnih delova fasade, potrebno je osim farbanja postaviti i traku po obodu svakog poligona i ofarbati površinu svakog poligona koji određuje oštećeni deo, naravno sve to ulazi u krajnju sumu cene. Ako je površina dela manja od 1m2 tada se koristi traka čija je cena 20 dinara po metru, a ako je površina veća koristi se skuplja traka čija cena iznosi 35 dinara po metru.

Cena farbe koja se koristi u svim slučajevima je 149,9 dinara po metru kvadratnom.



Teorijsko pojasnjenje kljucnih delova koda

U svrhu boljeg razumevanja koda potrebno je prvo malo bolje objasniti pojam poligon.

**Poligon** je figura u ravni koju čini mnogougaona linija i unutrašnja oblast određena tom linijom. Drugi naziv je *mnogougao*.

Ako sva temena mnogougla leže u jednoj [ravni](https://sr.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B2%D0%B0%D0%BD" \o "Ravan), mnogougao se naziva ravan mnogougao. To je mnogougao u užem smislu. Ako sva temena mnogougla ne leže u jednoj ravni, mnogougao se naziva [prostorni](https://sr.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%81%D1%82%D0%BE%D1%80" \o "Prostor) mnogougao. Duži koje čine mnogougaonu liniju nazivaju se **stranice mnogougla**. Temena izlomljene linije, krajevi stranica, nazivaju se **temena** mnogougla. Prema broju temena mnogougao je trougao, četvorougao, petougao, šestougao... Često se umesto mnogougla kaže i n-trougao (čita se entougao). Stranice mnogougla koje imaju zajedničko teme su susedne, a koje nemaju zajedničkih tačka su nesusedne. Ako je mnogougao [homeomorfan](https://sr.wikipedia.org/wiki/%D0%A5%D0%BE%D0%BC%D0%B5%D0%BE%D0%BC%D0%BE%D1%80%D1%84%D0%B8%D0%B7%D0%B0%D0%BC" \o ") kružnici, on se naziva prost mnogougao. Drugim rečima, prost mnogougao je mnogougao bez samopreseka, tj. kada:

1. iz svakog njegovog temena ishode samo dve stranice;
2. stranice nemaju zajedničkih tačaka (temena ne pripadaju stranicama);
3. temena ne leže na stranicama.

U [elementarnoj geometriji](https://sr.wikipedia.org/wiki/%D0%95%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%80%D0%BD%D0%B0_%D0%B3%D0%B5%D0%BE%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%98%D0%B0" \o "Elementarna geometrija) se najčešće posmatraju prosti mnogouglovi.

Matematicke formule za obim I povrsinu poligona su date sledecim izrazima:



**O=++…+**

Na osnovu ovih matematickih izraza mi smo napravili aplikaciju u programskom jeziku c#.

Sledeca stvar na koju treba obratiti paznju je forma same aplikacije koja je objektno orjentisana.

U okviru programskog koda imamo glavnu abstraktnu klasu “Fasada”.

Glavna klasa predstavlja roditeljsku klasu cije osobine nasledjuju potklase (deca klase). U nasem slucaju, potklase su: Delimicno I totalno ostecenje , a u okviru njih imamo I potklase poligon,teme I farba.

Abstraktna clasa(abstract class) je klasa koje ne sadrzi objekte, ali sadrzi abstraktne metode.

Abstraktne metode se koriste samo u okviru abstraktne klase I one ne sadrze telo, vec koriste telo nasledjene klase.

Kada je samo programiranje u pitanju postoje podaci koji ne bi trebali da budu dostupni svima. Da bi se to postiglo koriste se kljucne reci private-privatno, public-javno, protected-zasticeno.

Private-ovom tipu podataka se moze pristupiti kodom samo iz iste klase

Public-Tom tipu podataka se moze pristupiti bilo kojim kodom, bez obzira da li se nalazi u toj ili nekoj drugoj klasi.

Protected- tom tipu se pristupa bilo kojim kodom , ali u okviru dela gde je sam protected definisan

Enkapsulacija je jos jedan od osnovnih principa objektno orjentisanog programiranja. Ona podrazumeva skrivanje podataka od spoljnih pristupa. Primena enkapsulacije je veoma bitna pogotovu kod velikih projekata. Postovanjem pravila enkapsulacije, obezbedjujemo da objekti imaju strogo kontrolisane ulaze I izlaze, a samim tim smanjujemo mogucnost greske.

Svako pristupanje podacima vrsi se iskljucivo preko javnih metoda kreiranih za te potrebe-get I set.

U nasem slucaju- Dohvati I Postavi

Dohvati- vraca vrednost neke promenljive, dok metoda “postavi” sluzi za postavljanje neke nove vrednosti promenljive.

Zakljucak:

Objektno orjentisano nacin programiranja je postigao primat u modernom nacinu programiranja, zbog velike mogucnosti primene. Ovim projektnim zadatkom smo imali mogucnost to I da pokazemo, kao I da se na taj nacin posvetimo resavanju konkretnog problema.