

# - ZAVRŠNI PROJEKAT –

OBJEKTNO-ORIJENTISANO PROGRAMIRANJE

# DOKUMENTACIJA

**-TEMA PROJEKTA-**

Studenti: **Hamza Krkalić (359), Samra Smajić (407), Tea Tešnjak (355), Muhamed Mekić (403)**

Profesor: **Adnan Dželihodžić**

Asistenti: **Edin Tabak i Narcisa Hadžajlić**

Akademska godina: 2023 - 2024

1. **Osnovni opis projekta:**

Izabrali smo temu “Kozmetički salon” zbog toga što smo smatrali da je veoma opširna I izvodljiva tematika za projekat. Činila se dosta zanimljivom, a s time smo se složili dok smo pravili projekat. Ovaj projekat postavlja temelje za dalje usavršavanje i proširenje sistema u skladu s potrebama kozmetičkih salona. Kroz suočavanje s izazovima implementacije dodatnih funkcionalnosti i poboljšanja korisničkog interfejsa, studenti će razvijati dragocjeno iskustvo u programiranju, timskom radu i rješavanju problema.

Projekat za Kozmetički salon predstavlja korak ka razumevanju i primjeni programiranja u stvarnom poslovnom okruženju, posebno u oblasti ljepote i kozmetičke industrije. Kroz ovaj projekat, studenti stiču osnovna znanja programiranja i postavljaju temelje za dalje napredovanje u ovoj dinamičnoj oblasti.

1. **Funkcionalnosti**

* Klase:

KozmetickiSalon: Ova klasa predstavlja jezgro kozmetičkog salona, sa informacijama o uslugama i zaposlenima.

Atributi:

std::vector<Usluga> usluge: Vektor sa informacijama o različitim uslugama u salonu.

std::vector<Uposlenik> uposlenici: Vektor sa informacijama o zaposlenima u salonu.

Usluga: Klasa koja pruža informacije o pojedinačnoj usluzi u salonu.

Atributi:

std::string opisUsluge: Opis usluge.

double cijena: Cijena usluge.

std::string strucnjak: Stručnjak koji izvodi uslugu (može biti Manikir ili Pedikir).

Uposlenik: Klasa koja čuva podatke o zaposlenima u salonu.

Atributi:

double plata: Plata zaposlenika.

Pozicija pozicija: Enumeracija koja označava poziciju zaposlenika (Manikir, Pedikir, Recepcija).

bool zauzet: Indikator da li je zaposlenik trenutno zauzet.

* Enumeracija:

Pozicija: Enumeracija sa vrijednostima Manikir, Pedikir i Recepcija, označavajući različite pozicije u salonu.

* Pametni pokazivači:

Koristićemo pametne pokazivače (npr. std::shared\_ptr) za dinamičko upravljanje resursima.

* Virtuelne funkcije:

Virtuelne funkcije mogu se koristiti u klasama koje dijele zajedničko ponašanje, kao što su prikaziInformacije koje mogu biti prilagođene u podklasama.

* Preopterećeni operatori:

Preopterećivanje operatora (npr. << i >>) omogućava elegantan unos i ispis podataka, posebno korisno za rad sa uslugama.

Nasljeđivanje:

Klasa Uposlenik može naslijediti osnovne karakteristike iz klase Osoba, prateći princip nasljeđivanja i dijeljenja zajedničkih svojstava.

* Rad sa datotekama (radili smo na NTP):

Klasa će imati metode za efikasno čitanje i pisanje podataka u datotekama, kao što su ucitajUslugeIzFajla i sacuvajUslugeUFajl.

* Meni:

Implementacija metoda za korisnički meni, naprimjer prikaziGlavniMeni u klasi KozmetickiSalon, pruža jasan i interaktivan način za korisničku interakciju.

1. **GitHub repozitorij**

https://github.com/teatesnjak/Objektno-orijentisano-programiranje.git