

# - ZAVRŠNI PROJEKAT –

OBJEKTNO-ORIJENTISANO PROGRAMIRANJE

# DOKUMENTACIJA

**-Aplikacija za vremensku prognozu -**

Studenti: **Bejtić Kenan 331 , Buljić Imad 369, Fejzić Abdulah 360, Hinović Adi 345**

Profesor: **Adnan Dželihodžić**

Asistenti: **Edin Tabak i Narcisa Hadžajlić**

Akademska godina: 2023 - 2024

1. **Osnovni opis projekta:**

Naš projekat se zove "Aplikacija za vremensku prognozu".

Glavni razlog za izbor ove teme jeste njen veliki potencijal koji se nadovezuje na naše već stečeno znanje, ali u isto vrijeme pruža priliku da se istraže i nove mogućnosti koje nisu direktno povezane s našim redovnim školskim aktivnostima. Kroz ovaj projekt, naš tim je imao priliku da se upusti u rad s API-ima unutar C++ okruženja, što je bilo nešto sasvim novo za nas.

Kao studenti koji su se uglavnom fokusirali na "vanilla" C++ tokom školovanja, ovaj projekt nam je omogućio da istražimo kako moderni razvojni alati i okruženja, poput QT Creatora, mogu proširiti mogućnosti samog jezika. Ovo je bilo posebno uzbudljivo jer smo vidjeli kako napredne karakteristike nekih drugih programskih jezika i okruženja, poput onih koje nudi Visual Studio za C#, mogu biti primijenjene i unutar C++ ekosistema.

Pored tehničkog izazova i prilike da proširimo naše vještine, odabir teme "Aplikacija za vremensku prognozu" bio je i praktičan. Vremenska prognoza je nešto s čime se svakodnevno susrećemo i ima direktan utjecaj na naše živote. Kroz ovaj projekt, imali smo priliku ne samo da naučimo kako se razvija aplikacija za vremensku prognozu, već i da steknemo uvid u kompleksnost i važnost meteoroloških podataka i kako se oni obrađuju i prikazuju krajnjem korisniku.

Naš glavni cilj bio je stvoriti aplikaciju koja nije samo funkcionalna i pouzdana, već i intuitivna i laka za korištenje. Vjerujemo da smo kroz ovaj projekt uspjeli pokazati kako se teorijsko znanje stečeno na predavanjima može uspješno primijeniti u praksi, a istovremeno iskoristiti priliku za učenje novih, modernih tehnika programiranja koje nadilaze okvire tradicionalnog školskog rasporeda. Za samo kreiranje aplikacije , kao što je već spomenuto , koristili smo QT Creator koji je sa sobom donio nove mogućnosti koje se samo nadovezuju na već postojuće koje pruža C++.

1. **Funkcionalnosti**

Entiteti/Klase i Njihovi Atributi:

MainWindow: Glavna klasa aplikacije, koja nasljeđuje QMainWindow.

Sadrži različite UI elemente poput dugmadi, labela, scrollbara, itd.

Ključni atributi uključuju:

ui: Instanca korisničkog interfejsa.

api: Pametni pokazivač na WeatherApi klasu za pristup vremenskim podacima.

Različite labele i dugmadi za prikaz vremenskih podataka i interakciju s korisnikom.

WeatherApi: Klasa zadužena za komunikaciju s vremenskim API-jem. Atributi uključuju:

manager: Objekat za upravljanje mrežnim zahtjevima.

weather: Instanca klase WeatherDataALL za pohranu trenutnih vremenskih podataka.

currentRequestType: Enumeracija koja označava tip trenutnog zahtjeva.

WeatherDataALL: Klasa koja služi kao model za pohranu vremenskih podataka. Sadrži atribute poput temperature, vlažnosti, brzine vjetra, itd.

Međusobne Veze:

MainWindow i WeatherApi: MainWindow koristi WeatherApi za dohvat vremenskih podataka.

Ova veza se ostvaruje kroz pametni pokazivač api unutar MainWindow, demonstrirajući upotrebu pametnih pokazivača.

WeatherApi i WeatherDataALL: WeatherApi popunjava objekte klase WeatherDataALL sa podacima dobivenim iz API-ja. Ovo pokazuje kako se podaci iz jedne klase prenose i koriste u drugoj.

Dodatne Karakteristike:

Enumeracija: Klasa WeatherDataALL koristi enumeraciju WeatherCondition za definisanje različitih vremenskih uslova.

Pametni Pokazivači: Korišteni su za upravljanje instancama WeatherApi unutar MainWindow, što omogućava automatsko upravljanje resursima.

Iznimke: U WeatherApi klasi, iznimke se koriste za upravljanje greškama tokom mrežnih zahtjeva i obrade JSON podataka.

Preopterećeni Operatori: U klasi WeatherDataALL, operator << je preopterećen za ispisivanje objekata klase.

Nasljeđivanje: MainWindow nasljeđuje QMainWindow, demonstrirajući osnovni koncept OO programiranja.

Rad sa Datotekama: Implementirano u MainWindow za čitanje i pisanje favoritnih gradova.

Meni: UI MainWindow klase sadrži meni elemente za navigaciju i interakciju sa korisnikom.

Svaki od ovih elemenata ključan je dio aplikacije, demonstrirajući kako se različiti koncepti objektno orijentisanog programiranja i napredne funkcionalnosti C++ jezika mogu uspješno primijeniti u stvarnom projektu.

Uz to, odabir ove teme nam je omogućio da primijenimo različite aspekte programskog jezika C++, kao što su pametni pokazivači, iznimke, virtuelne funkcije i preopterećeni operatori, čime smo dobili bolje razumijevanje kako ovi koncepti funkcioniraju u praksi.

Qt Designer je dragocjen alat koji smo koristili za vizualno kreiranje korisničkog interfejsa naše aplikacije. S njim smo mogli lako postaviti widgete kao što su dugmad, labeli, scroll barovi i drop-down meniji. Designer nam je omogućio da vidimo kako će izgledati UI dok radimo na njemu, što nam je pomoglo da stvorimo čist i korisnički privlačan interfejs.

Integracija UI-a s Kodom: Nakon što smo vizualno kreirali UI, Qt Designer generira .ui fajl, koji se zatim može integrisati u naš C++ kod. U našem projektu, mainwindow.ui fajl sadrži definicije svih UI elemenata koje smo koristili u MainWindow klasi. Korištenjem Qt-ovog UIC-a (User Interface Compiler), ovaj .ui fajl se pretvara u C++ kod, koji se onda može koristiti unutar našeg aplikacijskog koda. Stilizacija UI-a: Koristili smo CSS-like stilove da dodamo estetski izgled našem UI-u. Qt podržava širok spektar stilskih opcija koje omogućavaju prilagođavanje izgleda widgeta, poput boja, fontova, granica itd. Ovo nam je omogućilo da našu aplikaciju učinimo vizualno privlačnijom i da poboljšamo korisničko iskustvo.

UI Interakcije Obrada Događaja: Interakcija korisnika s UI-om obradili smo kroz različite slotove i signal mehanizme koje Qt nudi. Na primjer, kada korisnik klikne na dugme za dodavanje grada u favorite, signal se emituje, a odgovarajući slot unutar MainWindow klase se aktivira da obradi taj događaj.

Dinamički UI Elementi: Za prikaz vremenskih podataka koristili smo dinamičke UI elemente. Na primjer, prognoza za sedam dana se prikazuje koristeći grid layout koji se popunjava programski, na osnovu podataka dobijenih iz WeatherApi klase. Kroz ovaj detaljan pristup dizajniranju korisničkog interfejsa, uspjeli smo stvoriti aplikaciju koja ne samo da je funkcionalna i informativna, već je i estetski privlačna i jednostavna za korištenje. Na ovaj način, naša aplikacija "Vremenska Prognoza" ne samo da zadovoljava osnovne funkcionalne zahtjeve, već pruža i ugodno korisničko iskustvo, što je jednako važno.

1. **Dodatno**

Izbor Radnog Okruženja - Qt Creator: Jedan od najznačajnijih kreativnih dodataka našem projektu je odluka da koristimo Qt Creator umjesto standardnog razvojnog okruženja poput Cliona. Ovaj izbor je bio motiviran željom da se izdvojimo od tradicionalnog pristupa C++ programiranju i da istražimo kako moderno IDE može poboljšati proces razvoja i kvalitetu konačnog proizvoda.

Qt Creator nudi integrirano okruženje koje olakšava razvoj GUI aplikacija, što je bilo ključno za stvaranje interaktivne i vizualno privlačne vremenske prognoze. Implementacija Grafičkog Korisničkog Interfejsa (GUI): Umjesto konvencionalnog terminalskog interfejsa, odabrali smo izraditi punokrvni GUI, što je dodalo sloj složenosti i kreativnosti našem projektu.

Koristeći Qt Designer, kreirali smo intuitivan i estetski ugodan interfejs koji omogućuje korisnicima lako praćenje vremenskih uslova, uz minimalan napor. Ovaj pristup je uvelike poboljšao korisničko iskustvo i funkcionalnost aplikacije. Integracija s Vremenskim API-em: Umjesto da se oslanjamo isključivo na statičke podatke, odlučili smo integrirati stvarne vremenske podat ke kroz API.

Ovo je bio značajan korak izvan osnovnih zahtjeva projekta. Korištenjem OpenWeatherMap API-a, aplikacija prikuplja stvarne podatke o vremenskim uslovima, što našoj aplikaciji daje dodatnu vrijednost i relevantnost. Ova integracija nije bila samo tehnički izazov, već je pružila i realnu primjenu podataka u stvarnom vremenu.

Napredna Obrada i Prikaz Podataka: Kreativno smo pristupili obr adi i prikazu podataka preuzetih sa API-a, transformišući sirove podatke u informativne i lako čitljive vizualne elemente. Na primjer, razvili smo funkcionalnost koja prikazuje vremensku prognozu po satima i danima, uključujući ikone vremenskih uslova, temperature, i vjetra.

Ovo je omogućilo korisnicima da na brz i efektivan način dobiju pregled vremenskih uslova, što je znatno unaprijedilo korisničko iskustvo.

Personalizacija Korisničkog Iskustva: Dodali smo mogućnost personalizacije aplikacije, omogućavajući korisnicima da dodaju i čuvaju svoje favorite gradove. Ova funkcionalnost omogućava korisnicima da brzo pristupe vremenskim uslovima za lokacije koje ih najviše interesuju, što aplikaciju čini mnogo praktičnijom i personalizovanijom.

1. **GitHub repozitorij**

https://github.com/AdiH16/Aplikacija-Za-Vremensku-Prognozu-OOP