Synchrotek
Science on board

PZ001

PROJEKTNI ZADATAK

Basic Python Kurs
Verzija 1.2

CONFIDENTIAL

Istorijat Dokumenta

Verzija	1.0	**Datum**	2023-10-31	**Autor**	Razni autori

- Inicijalna verzija

Verzija	1.1	**Datum**	2023-11-15	**Autor**	Razni autori

- Manje dopune u formalnoj postavci projektnog zadatka

Verzija	1.2	**Datum**	2023-11-17	**Autor**	Razni autori

- Dopuna dodatnih zahteva u vidu jasnog definisanja tipova automobila

Sadržaj

1. Uvod

Cilj ovog projektnog zadatka je pokazati stečeno znanje prilikom pohađanja „Basic Python“ kursa. Ovaj projekat će obuhvatiti sve ključne oblasti koje su bile predmet kursa, s posebnim fokusom na onim aspektima koji su od suštinskog značaja za buduće profesionalno usavršavanje i razvoj.

Tema ovog projektnog zadatka je "Sistem za Upravljanje Automobilima". Cilj projekta je razviti funkcionalni softverski sistem za vođenje automobilskog salona, omogućujući praćenje inventara automobila, upravljanje prodajom vozila, te generisanje potrebnih izlaznih datoteka za efikasno praćenje stanja i prometa svih transakcija u salonu. Projekat će obuhvatiti implementaciju različitih funkcionalnosti, uključujući evidenciju vozila, praćenje prodaje i generisanje. Ovo će omogućiti efikasno upravljanje automobilskim salonom i olakšati praćenje svih relevantnih podataka i poslovnih operacija.

Korisnik ovog sistema predstavlja prodavca u salonu automobila, i potrebno je napraviti sistem koji će ovaj korisnik s lakoćom moći da koristi, što znači da sistem mora razumljivo i jasno prikazivati sve podatke.

2. Korisnički zahtevi

U nastavku će biti opisani osnovni korisnički zahtevi koje sistem mora da ispunjava kroz realizaciju u programskom jeziku Python.

2.1 Unos novih automobila

Sistem omogućava korisnicima unos podataka o novim automobilima, uključujući *marku, model, godinu proizvodnje* i *cenu*. Korisnicima je dozvoljen unos više automobila odjednom.

Sistem vrši proveru unosa i osigurava da su svi podaci ispravno uneseni. Ukoliko korisnik nema dovoljno novca da izvrši uvoz nekog automobila na stanje, ispisati adekvatnu poruku.

2.2 Pretraga automobila po ceni

Korisnici mogu pretraživati dostupne automobile po rasponu cena, pružajući *minimalnu* i *maksimalnu cenu* vozila.

Rezultati pretrage treba da budu prikazani u obliku liste informacija o vozilima i na taj način ispisani korisniku. Cilj je korisniku prikazati pregledno sve informacije o automobilima po kriterijumima pretrage.

2.3 Prodaja automobila

Korisnici imaju mogućnost da izvrše prodaju navedenog automobila kupcu. Prilikom prodaje, unose se podaci o *kupcu* i *datumu prodaje*. Količina dostupnih automobila se ažurira nakon prodaje automobila, i dodatno treba pratiti izvedene transakcije tekućeg korisnika i njegov saldo.

2.4 Generisanje izveštaja

Sistem omogućava generisanje izveštaja o prodaji u PDF formatu. Izveštaj treba da sadrži detalje o svim transakcijama, uključujući *marku, model, cenu, datum prodaje* i *podatke o kupcu*.

2.5 Čuvanje istorije transakcija

Sistem treba da čuva istoriju svih transakcija u posebnoj datoteci. Svaka transakcija treba biti zabeležena u formatu koji sadrži *marku, model, cenu, kupca* i *datuma prodaje*.

Neophodno je čuvati ove informacije u formatu pogodnom za kasniju analizu.

2.6 Učitavanje podataka iz datoteke

Omogućiti učitavanje podataka o automobilima iz datoteke čiji je naziv uneo korisnik sistema. Podaci u datoteci treba da sadrže sledeće podatke: *marka, model, godina proizvodnje, cena*.

2.7 Organizacija funkcionalnosti u module

Neophodno je razdvojiti funkcionalnosti sistema u različite module, radi lakše organizacije koda i lakšeg proširenja i nadogradnji sistema. Bitno je definisati srodne celine po funkcionalnostima, poput celine za generisanje izveštaja ili celine za upravljanje automobilima.

2.8 Korisnički interfejs

Razviti osnovni korisnički interfejs za komunikaciju sa korisnikom kroz konzolu. Omogućiti unos podataka i čitanje podataka iz konzole.

3. Formalna postavka i klasni opis sistema

U narednom poglavlju će biti pružen opis svih ključnih klasa u sistemu, ističući njihove karakteristike, zajedno sa razmatranjem očekivanih ulaza i izlaza koje će svaka klasa obrađivati. Ovo će omogućiti dublje razumevanje arhitekture sistema i kako klase međusobno interaguju radi ostvarivanja funkcionalnosti sistema.

Bitno je zapaziti da je data samo skica sistema. Na realizatoru je da navedene klase dopuni novim metodama i/ili konstruktorima po potrebi, kao i da koristi atribute u klasi po potrebi, ali i realizovati sistem u objektnom duhu. Dodatno, moguće je dodati nove klase ukoliko se smatra neophodnim.

Moguće je menjati sve navedene klase, proširivati ih po potrebi ili dodavati nove argumente kako bi rešenje bilo kompletnije.

3.1 Klasa „Car“

- Konstruktor koji prima vrednosti potrebnih za kreiranje automobila koji se prate prema korisničkim zahtevima.
- Metoda „`__str__()`“ za formatiranje ispisa u konzolnom prikazu.

3.2 Klasa „CarManager“

- Metoda „`add_car(car)`“ koja dodaje novi automobil u sistem. Neophodno je proveriti da li se zaista dodaje objekat tipa klase *Car*.
- Metoda „`remove_car(car)`“ koja briše navedeni automobil iz sistema. Neophodno je proveriti da li se zaista briše objekat tipa klase *Car*.
- Metoda „`search_by_price(min_price, max_price)`“ koja omogućava pretragu zapamćenih automobila u opsegu [*min_price, max_price*] i vraća listu automobila koji ispunjavaju kriterijum.
- Metoda „`get_cars_list()`“ koja vraća celu listu automobila korisniku.

3.3 Klasa „User“

- Metoda „`sell_car(car, buyer)`“ koja omogućava prodaju automobila navedenom imenu kupca. Potrebno je adekvatno ažurirati listu dostupnih automobila (*CarManager*), listu transakcija korisnika prema navedenim korisničkim zahtevima i ažurirati saldo korisnika ovog sistema.
- Metoda „`add_new_car(make, model, year, price)`“ koja će praviti novi automobil (*Car*) sa navedenim atributima dodavati ga u inventar automobila (*CarManager*), a zatim ispisivati na konzoli poruku da li je uspešno dodat. Provizija korisnika sistema je 20% po

automobilu. Ukoliko korisnik nema dovoljno sredstava da izvrši unos automobila, ispisati adekvatnu poruku. Adekvatno ažurirati listu transakcija i dostupnih automobila.

- Metoda „`select_car(min_price, max_price)`“ koja vraća listu automobila čija je cena u navedenom intervalu. Korisnik kasnije može da odabere neki od automobila koji je vraćen u listi.
- Metoda „`import_from_file(file_name)`“ koja dodaje automobile navedene u JSON fajlu u sistem. Potrebno ažurirati liste transakcija i listu dostupnih automobila.
- Metoda „`generate_report()`“ pruža mogućnost generisanja izveštaja u PDF formatu* uz praćenje podataka definisanim u korisničkim zahtevima.
- Metoda „`save_transactions_to_file(file_name)`“ omogućava čuvanje transakcija u navedenoj CSV datoteci. Potrebno je pratiti koji je automobil u pitanju, da li je kupljen ili prodat i navesti jedinstveni identifikator automobila.

* Koristite `pip` za instaliranje eksterne biblioteke za generisanje PDF dokumenata za izveštaje o prodaji.

4. Dodatni zahtevi

U navedenom poglavlju će biti izložena lista dodatnih zahteva. Pred realizatorom sistema je izbor da li će njegov sistem da ih podrži ili ne. Ovi zahtevi upotpunjuju sistem i pružaju dodatne olakšice u njegovom korišćenju.

1. Podržati ispis izveštaja transakciju u dodatnom formatu (XML) pored već podržanog CSV formata.
2. Dodati mogućnost da sistem podrži i pretragu automobila prema godini proizvodnje.
3. Omogućiti da korisnik (*User*) unese novi automobil preko komandne linije.
4. Omogućiti praćenje tipa automobila (potrebno podržati: *limuzina, karavan, kabriolet*) pored osnovnih atributa.
5. Omogućiti pretragu automobila po tipu automobila.

5. Zaključak

Realizator je dužan da prati i da u velikoj meri ispoštuje što veći broj smernica navedenih tokom „Basic Python“ kursa, i da ispuni sve korisničke zahteve iz prvog poglavlja. Neophodno je pisati dokumentaciju i izvorni kod na engleskom jeziku, shodno prihvaćenoj konvenciji.

Realizatoru sistema su na raspolaganju sva dokumenta i materijali dostavljeni prilikom održavanja navedenog kursa, kao i dodatni materijali koji se mogu naći na drugim izvorima (internet, drugi kursevi...). Za sva dodatna pitanja, realizator može da se obrati predavačima „Basic Python“ kursa.