

UNIVERZITET U ZENICI
POLITEHNIČKI FAKULTET



ASP
ALGORITMI I STRUKTURE PODATAKA
2019/2020

ZADAĆA - Z1

Z1 -> DO POVEZANE LISTE

NAPOMENE:

- **Rok** za završetak zadaće je srijeda, **24.11.2019**
- Zadaću je potrebno poslati na email **sanid@netakademija.ba** najkasnije **do 23:59**
- Studenti su dužni zadaću uraditi **samostalno**
- **Izrada** zadaće je **obavezna**
- **Slanje** zadaće je **obavezno**
- **Zadaće poslane nakon isteka vremenskog roka neće biti pregledane**
- Studenti koji ne urade zadaću samostalno **neće ostvariti bodove -> 2.5 p**

Pri realizaciji ovih zadataka nije dozvoljeno uvoditi nove metode odnosno mijenjati način rada postojećih metoda niti korištenje bibliotečnih klasa (vector, deque, list...).

ZADATAK (RENT-A CAR):

Potrebno je napisati program koji će omogućiti slijedeće:

- Posuđivanje automobila iz rent-a-car agencije.
- Vraćanje automobila u agenciju
- Pregled sadržaja voznog parka
- Provjeru dostupnosti određenog automobila
- Pretragu voznog parka po proizvođaču automobila.
- Pregled svih automobila koje je određeni korisnik koristio
- Pregled svih korisnika agencije
- Dodavanje novog korisnika u agenciju

Ret-a-Car Agencija treba da bude implementirana koristeći **jednostruko povezanu listu**. U ovoj listi potrebno je da se nalaze automobili sa npr. sljedećim atributima: Naziv, Boja, Broj sjedišta, Proizvođač, Tip, Kategorija[Classic, Luksuzno, Teretno,...], Godina proizvodnje....

Lista proizvođača automobila također mora biti lista implementirana koristeći jednostruko povezanu listu. Implementirati klasu proizvođač koja će imati npr. slijedeće attribute (Naziv, Zelja porijekla, Ukupan broj dostupnih vozila...)

Kategorija vozila takođe mora biti lista implementirana koristeći jednostruko povezanu listu.



Za potrebe održavanja **baze korisnika** potrebno je koristiti jednostruko povezanu listu. Implementirati klasu korisnik čija su osnovna polja: Ime, Prezime, GodinaRođenja, Broj članske karte, Telefon, Email, BrojLoyaltyBodova

Osmisliti najbolji način za pamćenje automobila koje je korisnik posudio.

Napisati **glavni program** čiji je scenario:

1. Korisnik koji trenutno ne posjeduje člansku karticu dolazi u agenciju
2. Administrator dodaje korisnika u listu korisnika
3. Korisnik želi da pregleda sadržaj voznog parka
4. Taj korisnik želi provjeriti da li se neke automobil nalazi u voznom parku i da li je dostupna za iznajmljivanje
5. Korisnik, ukoliko je moguće, iznajmljuje automobil
6. Ukoliko nije moguće iznajmiti automobil, korisnik ima mogućnost da se izjasni da li želi najam nekog drugog automobila
7. Ukoliko korisnik kaže "DA", potrebno je vratiti se na korak 4
8. Ukoliko korisnik kaže "NE", usluga trenutnog korisnika završava, i administrator nakon njega može usluživati narednog korisnika
9. U biblioteku dolazi korisnik koji posjeduje člansku karticu i ranije je koristio usluge agencije
10. Korisnik želi da iznajmi automobil
11. Nakon uspješnog najma automobila korisnik odlazi
12. Korisnik iz koraka 9 vraća automobil koji je posudio u koracima 10 i 11
13. Svaki put kada korisnik uspješno vrati iznajmljeni automobil, računaju mu se Bodovi po formuli: $Cijena * Broj\ dana / 100$ i akumuliraju u varijablu BrojLoyaltyBodova

Program treba da radi tako da stalno "čeka" na administratorovu akciju. Obezbijediti način da administrator zaustavi rad programa (čime bi se simulirao kraj radnog vremena agencije).

Prilikom pisanja koda dozvoljeno je korištenje isključivo jednostruko povezane. Svako rješenje u kome se koristi neka druga struktura se neće priznati. Nije dozvoljeno korištenje bibliotečnih klasa (vector, deque, list...).

Svaka Klasa i Funkcija u nazivu treba da sadrže inicijale studenta na kraju npr. SM.

Svaka funkcija ispisa na ekrana završava u potpisu sa:

.....
broj indexa
.....