# 1Strukture podataka i algoritmi

## Projektni zadatak 008\_CARS

Vaš zadatak je napisati program (konzolnu aplikaciju u C++) koji će korisniku omogućiti rad s podacima o automobilima. Program mora biti smisleno strukturiran i intuitivan. U nastavku su opisane funkcionalnosti koje treba program imati. Za izradu projekta se koriste tri datoteke naziva **SPA\_PROJ\_008\_CARS\_data\_*n*.csv**. Opis stupaca:

* SPA\_PROJ\_008\_CARS\_data\_1.csv:
  + id: jedinstveni identifikator proizvođača.
  + name: naziv proizvođača.
  + short\_url: ne treba koristiti.
* SPA\_PROJ\_008\_CARS\_data\_2.csv:
  + id: jedinstveni identifikator modela.
  + brand\_id: identifikator proizvođača (veza na id iz SPA\_PROJ\_008\_CARS\_data\_1.csv).
  + name: naziv modela.
* SPA\_PROJ\_008\_CARS\_data\_3.csv:
  + id: jedinstveni identifikator verzije modela.
  + brand\_id: identifikator proizvođača (veza na id iz SPA\_PROJ\_008\_CARS\_data\_1.csv).
  + model\_id: identifikator modela (veza na id iz SPA\_PROJ\_008\_CARS\_data\_2.csv).
  + name: naziv verzije modela.
  + year: godina kad je verzija izdana.
  + price: cijena verzije u dolarima (ako je 0, cijena nije poznata).

### Ishod 1 (bodova: 20)

1. (Bodova: **1**) Ispišite naslov programa koristeći ASCII art, primjerice, možete uzeti nešto s <http://patorjk.com/software/taag> (ova funkcionalnost se svodi na jednostavan ispis red po red). Moguć izgled:---------------

\_\_\_\_\_\_\_ \_\_ \_\_\_\_\_\_ \_\_

\_\_\_\_\_\_\_\_/\\\\\\\\\\_\_\_\_\_/\\\\\\\\\\_\_\_\_\_\_\_/\\\\\\\\\\_\_\_\_\_\_\_\_\_/\\\\\\\\\\\\_\_\_

\_\_\_\_\_/\\\////////\_\_\_\_/\\\\\\\\\\\\\\_\_\_/\\\///////\\\\_\_\_\_\_/\\\/////////\\\\_

\_\_\_/\\\/\_\_\_\_\_\_\_\_\_\_\_\_/\\\/////////\\\\_\/\\\\_\_\_\_\_\/\\\\_\_\_\_\//\\\\_\_\_\_\_\_\///\_\_

\_\_/\\\\_\_\_\_\_\_\_\_\_\_\_\_\_\/\\\\_\_\_\_\_\_\_\/\\\\_\/\\\\\\\\\\\/\_\_\_\_\_\_\////\\\\_\_\_\_\_\_\_\_\_

\_\/\\\\_\_\_\_\_\_\_\_\_\_\_\_\_\/\\\\\\\\\\\\\\\\_\/\\\//////\\\\_\_\_\_\_\_\_\_\_\////\\\\_\_\_\_\_\_

\_\//\\\\_\_\_\_\_\_\_\_\_\_\_\_\/\\\/////////\\\\_\/\\\\_\_\_\_\//\\\\_\_\_\_\_\_\_\_\_\_\_\////\\\\_\_\_

\_\_\///\\\\_\_\_\_\_\_\_\_\_\_\/\\\\_\_\_\_\_\_\_\/\\\\_\/\\\\_\_\_\_\_\//\\\\_\_\_/\\\\_\_\_\_\_\_\//\\\\_\_

\_\_\_\_\////\\\\\\\\\\_\/\\\\_\_\_\_\_\_\_\/\\\\_\/\\\\_\_\_\_\_\_\//\\\\_\///\\\\\\\\\\\/\_\_\_

\_\_\_\_\_\_\_\/////////\_\_\///\_\_\_\_\_\_\_\_\///\_\_\///\_\_\_\_\_\_\_\_\///\_\_\_\_\///////////\_\_\_\_\_

1. (Bodova: **2**) Dizajnirajte i implementirajte tip podataka CarVersion koji može čuvati sve podatke o verziji automobila: naziv verzije, naziv proizvođača, naziv modela, godina izdanja, cijena. Pripremite tip podataka tako da se može konstruirati na smislene načine te definirajte gettere, settere i potrebne metode za svakog člana. Omogućite korisniku da pozivanjem metode na objektu može ispisati osnovne podatke o automobilu na ekran (naziv verije, naziv proizvođača, naziv modela). ----------------
2. (Bodova: **7**) Prikažite korisniku opciju u izborniku „Unos podataka iz datoteke“. Kad korisnik odabere opciju, podatke iz sve tri datoteke (neka se nalaze u istom folderu kao i projekt) isparsirajte u vektor. Taj vektor sad predstavlja vašu bazu podataka automobila i osnova je za sve ostale zadatke. ----------------
3. (Bodova: **2**) Izmjerite i ispišite koliko je trajalo učitavanje datoteke u vektor. ----------------
4. (Bodova: **4**) Ispišite korisniku top ljestvicu proizvođača prema broju verzija modela, od najvećeg broja prema najmanjem. ----------------
5. (Bodova: **4**) Poneka verzija sadrži i podatak o volumenu motora u obliku dva broja odvojena točkom. Primjerice, „Alfa Romeo MiTo 1.4 T MultiAir Quadrifoglio Verde“ kaže da je volumen motora 1.4. Ispišite korisniku top pet volumena. Volumen ćete prepoznati po razmaku, zatim jednoj brojci, zatim točki, zatim drugoj brojci i na kraju opet razmaku.

### Ishod 2 (bodova: 20)

1. (Bodova: **3**) Prikažite korisniku opciju u izborniku „Kopiranje datoteke“. Kad korisnik odabere opciju, pitajte ga koju datoteku želi kopirati te želi li prepisati retke iz datoteke u istom ili obrnutom redoslijedu. Prikladno koristeći stog ili red, prepišite retke u željenom redoslijedu iz željene datoteke u odredišnu datoteku u istom folderu. ----------------

0

1. (Bodova: **3**) Prikažite korisniku opciju u izborniku „Obrada po proizvođačima“. Kad korisnik odabere opciju, napravite sljedeće operacije:

* Pitajte korisnika da unese naziv proizvođača kojeg želi obraditi.
* Prekopirajte sve verzije traženog proizvođača u novu listu (koristite list<T>).
* Obrišite sve verzije iz liste čija cijena nije poznata.
* Poslažite preostale verzije u listi obrnutim redoslijedom (prva postaje zadnja, druga predzadnja, …)
* Ispišite verzije iz liste. ----------------

1. (Bodova: **4**) Koristeći iteratore, krenite od zadnje verzije u vektoru prema prvoj i:

* Svaki put kad je stog prazan, stavite sljedeći objekt na njega.
* Svaki put kad naiđete na objekt *x* koji ima model jednak modelu na vrhu stoga, dodajte *x* na stog.
* Svaki put kad naiđete na objekt *x* koji ima model različit modelu na vrhu stoga, ispraznite stog i dodajte objekt *x* na stog.

Na kraju ispišite sve što je preostalo na stogu.

1. (Bodova: **5**) Prikažite korisniku statistiku verzija. Kad korisnik odabere opciju, prikažite u svakom retku po jedan model i pokraj njega broj njegovih verzija (neka # označava najviše 10 verzija). Primjer početka prikaza (modeli i brojevi su izmišljeni): ----------------

MiTo # (2)

Giulietta ### (30)

Challenger ## (17)

1. (Bodova: **5**) Omogućite korisniku zadavanje naredbi kojima prikazuje željene verzije u vektoru. Postupite ovako:

* Kad korisnik zada naredbu „SELECT \* FROM Cars WHERE Volume = '*n*'“, prikažite mu traženi zapis. Neka je n traženi volumen motora, primjerice, 1.6.
* Za sve ostale naredbe ispišite mu „Naredba nije podrzana“.