# Strukture podataka i algoritmi

## Projektni zadatak 009\_FLIGHTS

Vaš zadatak je napisati program (konzolnu aplikaciju u C++) koji će korisniku omogućiti rad s podacima o letovima. 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\_009\_FLIGHTS\_data\_*n*.csv**. Opis stupaca koji su vam potrebni (ostale nije potrebno koristiti):

* SPA\_PROJ\_009\_FLIGHTS\_data\_1.csv:
  + IATA\_CODE: Airline Identifier (potreban kao veza za "SPA\_PROJ\_009\_FLIGHTS\_data\_3.csv" - AIRLINE)
  + AIRLINE: Airline's Name
* SPA\_PROJ\_009\_FLIGHTS\_data\_2.csv:
  + IATA\_CODE: Airport Identifier (potreban kao veza za "SPA\_PROJ\_009\_FLIGHTS\_data\_3.csv" - ORIGIN\_AIRPORT i DESTINATION\_AIRPORT)
  + AIRPORT: Airport's Name
  + CITY: City Name of the Airport
  + STATE: State Name of the Airport
  + COUNTRY: Country Name of the Airport
* SPA\_PROJ\_009\_FLIGHTS\_data\_3.csv (zbog veličine, datoteka je zipana. Odzipajte je prije korištenja):
  + YEAR: Year of the Flight Trip
  + MONTH: Month of the Flight Trip
  + DAY: Day of the Flight Trip
  + DAY\_OF\_WEEK: Day of week of the Flight Trip
  + AIRLINE: Airline Identifier (veza prema "SPA\_PROJ\_009\_FLIGHTS\_data\_1.csv" - IATA\_CODE)
  + FLIGHT\_NUMBER: Flight Identifier
  + TAIL\_NUMBER: Aircraft Identifier
  + ORIGIN\_AIRPORT: Starting Airport (veza prema "SPA\_PROJ\_009\_FLIGHTS\_data\_2.csv" - IATA\_CODE)
  + DESTINATION\_AIRPORT: Destination Airport (veza prema "SPA\_PROJ\_009\_FLIGHTS\_data\_2.csv" - IATA\_CODE)
  + SCHEDULED\_DEPARTURE: Planned Departure Time
  + DEPARTURE\_TIME: Actual Departure Time
  + DEPARTURE\_DELAY: Total Delay on Departure
  + CANCELLED: Flight Cancelled (1 = cancelled)
  + CANCELLATION\_REASON: Reason for Cancellation of flight: A - Airline/Carrier; B - Weather; C - National Air System; D – Security

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 FlightDetails koji može čuvati sve podatke o letovima iz datoteke **SPA\_PROJ\_009\_FLIGHTS\_data\_3.csv**. 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 letu na ekran (datum, kompanija i broj leta, polazišni aerodrom, odredišni aerodrom). Primjerice, za neki let treba ispisati: 1.1.2015, AA2336, LAX ==> PBI.
2. (Bodova: **2**) Prikažite korisniku opciju u izborniku „Unos mjeseca i godine“. Kad korisnik odabere opciju, pitajte ga da upiše mjesec i godinu pa upisane podatke spremite u binarnu datoteku.
3. (Bodova: **4**) Prikažite korisniku opciju u izborniku „Unos podataka iz datoteke“. Kad korisnik odabere opciju, otvorite datoteku **SPA\_PROJ\_003\_HEROES\_data\_3.csv** te sav sadržaj iz nje za zadani mjesec i godinu (pročitajte ih iz binarne datoteke) isparsirajte u vektor. Taj vektor sad predstavlja vašu bazu podataka letova i osnova je za sve ostale zadatke.
4. (Bodova: **2**) Izmjerite i ispišite koliko je trajalo učitavanje datoteke u vektor.
5. (Bodova: **5**) Promijenite tip podataka FlightDetails tako da osim kratica sadrži i pune nazive za kompaniju, polazišni i odredišni aerodrom (imate ih u preostale dvije datoteke). Promijenite ispis leta tako da sad uključuje pune nazive. Promijenite postupak parsiranja.
6. (Bodova: **4**) Ispišite top deset kompanija prema broju letova u vektoru.
7. (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.
8. (Bodova: **3**) Prikažite korisniku opciju u izborniku „Obrada po kompanijama“. Kad korisnik odabere opciju, napravite sljedeće operacije:

* Pitajte korisnika da unese kompaniju koju želi obraditi.
* Prekopirajte sve letove tražene kompanije u novu listu (koristite list<T>).
* U petlji pitajte korisnika da unese polazišni ili odredišni aerodrom i njegov naziv, a zatim obrišite sve takve letove iz liste.
* Ispišite preostale letove iz liste.

1. (Bodova: **4**) Koristeći iteratore, krenite od zadnjeg leta u vektoru prema prvom i:

* Gurnite na stog broj 0.
* Kad naiđete na otkazani let, preskočite ga.
* Kad naiđete na neotkazani let, skinite trenutni broj sa stoga, pribrojite mu DEPARTURE\_DELAY trenutnog leta i dobiveni broj ponovno stavite na stog.

Na kraju ispišite broj sa stoga.

1. (Bodova: **5**) Prikažite korisniku statistiku otkazanih letova. Kad korisnik odabere opciju, prikažite u svakom retku po jednu kompaniju model i pokraj nje broj otkazanih letova (neka # označava najviše 10 letova). Primjer početka prikaza (kompanije i brojevi su izmišljeni):

United Air Lines Inc. # (2)

American Airlines Inc. ### (30)

Hawaiian Airlines Inc. ## (17)

1. (Bodova: **5**) Omogućite korisniku zadavanje naredbi kojima prikazuje željene letove iz vektora:

* Kad korisnik zada naredbu „SELECT \* FROM Flights WHERE From = '*n1*' AND To = '*n2*'“, prikažite mu tražene zapise. Neka je *n1* traženi polazišni aerodrom, a *n2* traženi odredišni aerodrom. Pretraživanje napravite neovisno o velikim i malim slovima.
* Za sve ostale naredbe ispišite mu „Naredba nije podrzana“.