




PROGRAMIRANJE
Predavanje 14 – Tekstualne datoteke

Ishod učenja 6

1

TEKSTUALNE DATOTEKE

Strana • 2



2

Uvod

- Rad s tekstualnim datotekama je vrlo slično čitanju s tipkovnice i pisanju u konzolu
 - C++ tretira datoteke kao tijekove (engl. *streams*)
 - Sve što smo naučili o `cin` i `cout` vrijedi i za tekstualne datoteke
- Postoje dva tipa datotečnih tijekova (engl. *file streams*):
 - Ulazni datotečni tijek – **`ifstream`**
 - Donosi podatke iz datoteke u naš program
 - Izlazni datotečni tijek – **`ofstream`**
 - Odnosi podatke iz našeg programa u datoteku
- Prije nego možemo koristiti datotečni tijek, moramo uspostaviti vezu s konkretnom datotekom na disku (moramo „otvoriti datoteku”)

Strana 3



3

Otvaranje datoteke

- Objekti potrebni za rad s datotekama se nalaze u `fstream`:

```
#include <fstream>
```

- Da bismo otvorili datoteku, moramo:

- Kreirati objekt tipa `ifstream` ili `ofstream`
 - Ovisno želimo li čitati ili pisati podatke
- Navesti putanju do datoteke u okruglim zagradama
 - Tu zapravo pozivamo konstruktorsku funkciju

```
ifstream dat1("D:\\temp\\file1.txt");
ofstream dat2("file2.txt");
```

Pozivanje konstruktorske funkcije

Strana 4



4

Provjera uspjeha

- Otvaranje datoteke može biti neuspješno iz niza razloga:
 - Putanja pokazuje na datoteku koja je postoji
 - Datoteku je već netko otvorio u ekskluzivnom načinu
 - Nemamo prava pisanja u direktorij u kojem je datoteka
- Standardni način provjere uspješnosti u main-u:


```
if (!dat1) {
    cout << "Pogreska kod otvaranja datoteke" << endl;
    return 1;
}
```
- Ako otvaranje ne uspije, varijabla `dat1` "sadrži" `false`

Strana • 5



5

Zatvaranje datoteke

- Otvorena datoteka koristi resurse operacijskog sustava
 - Pravilo kod programiranja: resurse operacijskog sustava **uzmite što kasnije** možete i **otпустite što ranije** možete
- Oba tipa datoteka zatvaramo na jedan od dva načina:
 - Eksplicitno, pozivanjem metode **close()** na tijeku:


```
dat1.close();
```
 - Implicitno, puštanjem da varijabla ode iz dosega (engl. *scope*)
 - Doseg lokalne varijable je funkcija u kojoj je deklarirana
 - Ispadanjem iz dosega poziva se destruktorska funkcija koja otpušta resurse
 - Mi ćemo uvijek eksplicitno zatvarati datoteku

Strana • 6



6

ČITANJE I PISANJE U TEKSTUALNU DATOTEKU

Strana • 7



7

Čitanje iz datoteke

- Prvi korak je napraviti ulazni datotečni tijek prema datoteci:
`ifstream dat1("Datoteka1.txt");`
- Moramo provjeriti je li otvaranje uspješno
- Nakon toga, čitanje radimo jednako kao iz cin:
`getline(dat1, first_ime); // Za stringove`
`dat1 >> n; // Za ugrađene druge tipove podataka`

Strana • 8



8

Čitanje iz datoteke

- Operacija čitanja vraća true or false
 - Ako smo stigli do kraja datoteke, vratit će false:


```
if (dat1 >> n) {
    // Čitanje je bilo uspješno...
}
```

```
if (getline(dat1, s)) {
    // Čitanje je bilo uspješno...
}
```
- Kad smo gotovi, moramo zatvoriti datotečni tijek:


```
dat1.close();
```

Strana • 9



9

Pisanje u datoteku

- Prvi korak je kreirati izlazni datotečni tijek prema datoteci (postojećoj ili nepostojećoj):


```
ofstream dat2("Datoteka2.txt");
```

 - Ako datoteka ne postoji, bit će kreirana
 - Ako datoteka postoji, bit će obrisana i kreirat će se nova, prazna
- Moramo provjeriti je li otvaranje uspješno
- Nakon toga, pisanje radimo jednako kao u cout:


```
dat2 << "Hello!" << endl;
```
- Kad smo gotovi, moramo zatvoriti datotečni tijek:


```
dat2.close();
```

Strana • 10



10

Primjeri

1. Napišite program koji od korisnika učitava 5 brojeva i upisuje ih u datoteku, svaki u svoj red.
2. Napišite program koji od korisnika učitava 5 brojeva i upisuje ih u datoteku obrnutim redoslijedom i odvojene zarezima. Iza zadnjeg broja nemojte staviti zarez.
3. Kreirajte polje s 5 osoba (ime, prezime, godina). Spremite ih u datoteku tako da svaka osoba ide u svoj vlastiti red s vrijednostima odvojenim tabovima.

Strana • 11



11

Primjeri

4. Napravite datoteku i u nju upišite 5 cijelih brojeva, svaki u svoj red. Napišite program koji korisniku ispisuje tih 5 brojeva.
5. Napravite datoteku i u nju upišite n cijelih brojeva, svaki u svoj red. Napišite program koji korisniku ispisuje sve brojeve.
6. Napravite datoteku i u nju upišite n cijelih brojeva, svaki u svoj red. Napišite program koji u novu datoteku upisuje sve parne brojeve iz prve datoteke, svaki u svoj red.
7. Učitajte naziv datoteke od korisnika i ispišite njen sadržaj.

Strana • 12



12

Primjeri

8. Napišite funkciju koja prima više stringova i radi sljedeće: za svaki primljeni string gleda broj znakova. Ako je manji od 5, upisuje string u datoteku "kratke_rijeci.txt", ako nije, upisuje ga u datoteku "duge_rijeci.txt". Napišite program koji od korisnika učitava onoliko riječi koliko korisnik želi, a nakon toga prosljeđuje riječi u funkciju.

Strana • 13



13

Zadaci za sljedećih 7 dana

- Prije sljedećeg predavanja trebate:
 1. Pročitati iz *Demistificirani C++*:
 - 21.7 Datotečni ispis i učitavanje

Strana • 14



14