

Datoteke



Formatirani način upisa i ispisa podataka

Planirani ishodi učenja:

- upisati i čitati podatke iz datoteke korištenjem naredbi **fscanf()** i **fprintf()** uz primjenu prikladnih opisa formata

Ponovimo...

- Datoteke koristimo za **trajnu pohranu** podataka.
- Rad s datotekama omogućuje pohranu rezultata programa ili rad s podacima iz datoteke. Za to je potrebno definirati **tok (stream)**.
- Biblioteka **fstream** nam omogućuje rad s datotekama, a u zaglavlju dokumenta potrebno je zadržati i **iostream**. U biblioteku **fstream** uključene su **metode** koje nam omogućuju upravljanje datotekama na disku uređaja:

Ponovimo...

ofstream

Omogućuje stvaranje datoteka i pisanje u njih

ifstream

Omogućuje čitanje iz datoteka

fstream

Kombinacija prethodnih metoda: čita i piše u datoteke

- **Funkcije** za rad s datotekama:

Ponovimo...

open()

Omogućuje stvaranje datoteke

read()

Omogućuje čitanje iz datoteke

write()

Omogućuje upis novih podataka

close()

Omogućuje zatvaranje datoteke

Ponovimo...

- Da bismo stvorili datoteku koristimo:
 - ili **ofstream** ili **fstream** objekt i
 - navodimo **ime objekta**
 - **ime** i **nastavak** (ekstenziju) **datoteke** koju želimo stvoriti.
- Za ispis u datoteku koristimo operator ispisa **<<**
- Po završetku rada s datotekom zatvaramo tok prema datoteci uporabom funkcije **close()**

Ponovimo...

- Za čitanje iz datoteke koristimo:
 - ili **ifstream** ili **fstream** objekt i
 - navodimo **ime objekta**
 - **ime** i **nastavak** (ekstenziju) **datoteke** iz koje želimo čitati.
- Upareno koristimo **while petlju** i funkciju **getline()** kako bismo čitali podatke iz datoteke redak po redak
- Za prijenos podataka iz datoteke u programski koristimo operator unosa **>>**
- Po završetku rada s datotekom zatvaramo tok prema datoteci uporabom funkcije **close()**

Ponovimo...

• Do sada smo čitali iz datoteka i pisali podatke u datoteke s nastavkom (ekstenzijom):

- **txt**
- **doc**
- **xls**
- **csv**

fstream

- Prilikom uporabe objekta **fstream** potrebno je ili **unaprijed stvoriti datoteku** u koju ćemo pisati ili treba postaviti **oznaku načina rada (Mode Flag)**

Oznaka načina rada (Mode Flag)	opis
ios::in	Otvaranje datoteke za čitanje (defaultno za ifstream)
ios::out	Otvaranje datoteke za pisanje (defaultno za ofstream)
ios::app	Otvaranje datoteke i dodavanje zapisa na kraj datoteke
ios::ate	Otvaranje datoteke za ispis i postavljanje mogućnosti čitanja/pisanja na kraj datoteke
ios::trunc	Otvaranje datoteke i brisanje njenog sadržaja prije zapisivanja novog

Primjer:

```
1  #include <iostream>
2  #include <fstream>
3
4  using namespace std;
5
6  int main()
7  {
8      fstream mojDokument("ispis.txt");
9      if (mojDokument.is_open()){
10         mojDokument<<"Neki probni tekst"<<endl;
11     }
12     else{
13         cout<<"Ne mogu otvoriti datoteku..."<<endl;
14     }
15     mojDokument.close();
16     return 0;
17 }
```

Ne mogu otvoriti datoteku...

Ako sami unaprijed otvorimo datoteku **ispis.txt** i postavimo je u mapu projekta onda ovaj programski kod radi bez greške:

Problem rješavamo postavljanjem **oznake načina rada (Mode Flag)**:

```
1  #include <iostream>
2  #include <fstream>
3
4  using namespace std;
5
6  int main()
7  {
8      fstream mojDokument;
9      mojDokument.open("ispis.txt", ios::out);
10     if (mojDokument.is_open()){
11         mojDokument<<"Neki probni tekst"<<endl;
12     }
13     else{
14         cout<<"Ne mogu otvoriti datoteku..."<<endl;
15     }
16     mojDokument.close();
17     return 0;
18 }
```

ispis.txt – Blok za pisanje

Datoteka Uređivanje Oblikovanje Prikaz Pomoć
Neki probni tekst

Oznake načina rada (Mode Flag) možemo i kombinirati uporabom | između:

```
1  #include <iostream>
2  #include <fstream>
3
4  using namespace std;
5
6  int main()
7  {
8      fstream mojDokument;
9      mojDokument.open("ispis.txt", ios::out|ios::app);
10     if (mojDokument.is_open()){
11         mojDokument<<"Neki probni tekst"<<endl;
12     }
13     else{
14         cout<<"Ne mogu otvoriti datoteku..."<<endl;
15     }
16     mojDokument.close();
17     return 0;
18 }
```

ispis.txt – Blok za pisanje

Datoteka Uređivanje Oblikovanje Prikaz Pomoć

Neki probni tekst

Neki probni tekst

Ponovimo...

Neformatirane datoteke
sadržijo samo podatke i
ništa više.



U neformatiranoj se
datoteci podaci zapisuju s
onoliko bitova koliko taj
tip podataka zauzima.

Formatirani način upisa i ispisa podataka

- Do sada samo za upis i ispis podataka uvijek koristili samo naredbe:

1. **cin**

2. **cout**

- One se koriste kada nam **format zapisa nije važan.**
- U radu s datotekama imamo potrebu za navođenjem formata zapisa.

Formatirani način upisa i ispisa podataka

- Za formatirani unos imamo funkcije:

1. scanf – služi za pohranu podataka u varijable, opći oblik joj je:

scanf ("opis_formata",[adresa, adresa..]);

u kodu to izgleda ovako:

```
scanf ("%d", &a);
```

Formatirani način upisa i ispisa podataka

- **Opis formata** se sastoji od oznake **postotka (%)** iza kojeg slijedi **slovo**, koje opisuje **tip podatka** koji se upotrebljava. Jedan ili više opisa formata je smješteno između dvostrukih navodnika, a najčešće korišteni opisi su:

Funkcija	scanf	printf
char	%c	%c
int	%d	%d
float (decimalni)	%f	%f
float (eksponencijalni)	%e	%e
string (niz znakova)	%s	%s
unsigned	%u	%u
long int	%ld	%ld
double	%lf	%lf
long double	%Lf	%Lf

Formatirani način upisa i ispisa podataka

2. **printf** pokazuje vrijednost varijabli na ekranu:

printf ("opis_formata", varijabla, varijabla,...);

```
printf("Rezultat je %d", a);
```

opis_formata može sadržavat tekst koji će se ispisati. Za ispis varijabli, potrebno je na odgovarajuće mjesto upisati opis formata pretvorbe za svaku varijablu. Opisi formata za funkciju **scanf** su isti kao za funkciju **printf**.

Za korištenje ovih funkcija potrebna nam je i biblioteka:

<stdio>

Formatirani način upisa i ispisa podataka

Sekvenca	Ime	Značenje
\a	Alert	Aktiviranje zujala
\b	Backspace	Vraćanje unatrag jedan znak
\f	Formated	Start nove strane ili ekrana
\n	Newline	Pomicanje u novu liniju
\r	Carriage ret	Povratak na početak linije
\t	Horizontal tab	Pomicanje na narednu tab poziciju
\v	Vertical tab	Pomicanje prema dolje
\\	Backslash	Prikazivanje kose crte
\'	Single quote	Prikazivanje jednostrukog navodnika
\"	Double quote	Prikazivanje dvostrukih navodnika
\?	Question mark	Prikazivanje znaka pitanja
\000		Prikazivanje znakova sa oktalno zadanim ASCII kodom
\xHHH		Prikazivanje znakova sa heksadecimalno zadanim ASCII kodom

Formatirani način upisa i ispisa podataka

1. primjer: Pomoću formatiranog unosa i ispisa izračunati zbroj i umnožak dva cijela broja.

```
1  #include <cstdlib> //zbog system("pause")
2  #include <iostream> //unos i ispis preko tipkovnice odnosno konzole
3  #include <cstdio> //zbog funkcija scanf i printf
4  using namespace std;
5  int main()
6  {
7      int a,b,c,d; //deklaracija 4 cjelobrojne varijable
8      printf ("Unesi prvi broj: ");
9      scanf ("%d",&a); //oznaka %d govori da ce varijabla biti cjelobrojna,
10     // a &a da ce biti pohranjena na adresi varijable a
11     printf("Unesi drugi broj: ");
12     scanf ("%d",&b); //oznaka %d govori da ce varijabla biti cjelobrojna,
13     // a &b da ce biti pohranjena na adresi varijable b
14     c=a+b;
15     d=a*b;
16     printf ("Umnozak je %d , a zbroj %d.\n",d,c); //za svaku varijablu prvo imamo tip,
17     //a zatim navodimo varijable koje treba ispisati redom kako je predvidjen ispis
18     system ("PAUSE");
19     return 0;
20 }
```

```
Unesi prvi broj: 4
Unesi drugi broj: 5
Umnozak je 20 , a zbroj 9.
```

Formatirani način upisa i ispisa podataka

2. primjer: Pomoću formatiranog unosa i ispisa izračunati zbroj i umnožak dva realna broja.

```
1  #include <cstdlib> //zbog system("pause")
2  #include <iostream> //unos i ispis preko tipkovnice odnosno konzole
3  #include <cstdio> //zbog funkcija scanf i printf
4  using namespace std;
5  int main()
6  {
7      float a,b,c,d; //deklaracija 4 realne varijable
8      printf ("Unesi prvi broj: ");
9      scanf("%f",&a); //oznaka %f govori da ce varijabla biti realna,
10     // a &a da ce biti pohranjena na adresi varijable a
11     printf("Unesi drugi broj: ");
12     scanf("%f",&b); //oznaka %f govori da ce varijabla biti realna,
13     // a &b da ce biti pohranjena na adresi varijable b
14     c=a+b;
15     d=a*b;
16     printf ("Umnozak je %f , a zbroj %f.\n",d,c); //za svaku varijablu prvo imamo tip,
17     //a zatim navodimo varijable koje treba ispisati redom kako je predvidjen ispis
18     system ("PAUSE");
19     return 0;
20 }
```

```
Unesi prvi broj: 4
Unesi drugi broj: 5
Umnozak je 20.000000 , a zbroj 9.000000.
```


Formatirani način upisa i ispisa podataka

- Želimo li koristiti formatirani upis/ispis podataka u datoteku koristiti ćemo naredbe:

1. **fscanf()**

2. **fprintf()**

Formatirani način upisa i ispisa podataka

- Naredba **fscanf()** čita podatke iz toka i pohranjuje ih u ovisnosti o formatima parametara na lokacije na koje pokazuju dodatni argumenti. Oni mogu pokazivati na postojeće objekte sa odgovarajućim formatima.

```
int fscanf ( FILE * stream, const char * format, ... );
```


Formatirani način upisa i ispisa podataka

- Naredba **fprintf()** upisuje podatke određenog formata u datoteku. Ukoliko format uključuje posebne znakove (koji počinju s %) ubacuju se i oni .

```
int fprintf ( FILE * stream, const char * format, ... );
```

Formatirani način upisa i ispisa podataka

- Prije otvaranja datoteke programu ćemo naredbom:

FILE *identifikator;

najaviti da ćemo raditi s datotekom.

- **Identifikator** je naziv datoteke koji ćemo koristiti u programu, a **zvjezdica** govori da se radi o **pokazivaču**.

Formatirani način upisa i ispisa podataka

- **FILE** je objekt koji sadrži informacije o kontroli toka.
- Obično se stvaraju pomoću **fopen** ili **tmpfile** a obje te funkcije vraćaju pokazivač na objekt.
- Za rad s ovom naredbom trebamo pozvati biblioteku **<stdio>**

Formatirani način upisa i ispisa podataka

- Datoteku ćemo otvoriti funkcijom:

`fopen();`

a nakon uporabe zatvoriti funkcijom:

`fclose();`

kako bismo oslobodili memorijski prostor.

- Općenito se te funkcije zapisuju:

`fopen("ime_datoteke","način_otvaranja"); fclose(identifikator);`

Formatirani način upisa i ispisa podataka

- Za pisanje u datoteku koristi se nekoliko funkcija. Najčešće korištene funkcije su:

1. **fprintf()**

2. **fwrite()** - pogodna za upisivanje **blokova podataka**.

Formatirani način upisa i ispisa podataka

- Za čitanje podataka iz datoteke najčešće se koriste funkcije:

1. **fscanf()**

2. **fread()** - koja se često koristi u kombinaciji sa **strukturama**.

Formatirani način upisa i ispisa podataka

U programima za rad s datotekama često ćemo naići na dio programskog koda kojim se **ispituje da li je datoteka ispravno otvorena**. Ako je datoteka deklarirana sa **FILE *f**; onda taj dio koda izgleda ovako:

```
if (f==NULL)
{
    printf("Greska pri otvaranju");
    exit(1);
}
```

Ako je uvjet **(f==NULL)** ispunjen na zaslonu računala će se ispisat poruka "Greška pri otvaranju".

Formatirani način upisa i ispisa podataka

- U formatiranu (neki ju nazivaju i tekstualna) datoteku se podaci zapisuju tako da se za svaki znak rezervira **8 bitova**, dakle svaki podatak - **slovo ili broj** (zapisan kao binarni ascii kod) se s lijeve strane nadopunjava nulama. Svaki znak ima **8 znakova** i na taj način recimo tekstualni editori znaju kako tretirati te podatke, da je svakih 8 bitova jedan znak zapisan ascii kodom.

Formatirani način upisa i ispisa podataka

3. primjer: Pomoću formatiranog ispisa, ispisati kvadrate brojeva od 1 do 10 u datoteku kvadrati.txt. Koristiti FILE, fopen, fclose i fprintf.

```
1  #include <iostream>
2  #include <cstdio>
3  #include <cstdlib>
4  using namespace std;
5
6  int main()
7  {
8      FILE*f;
9      f=fopen("kvadrati.xls","w");
10     if (f==NULL)
11     {
12         printf("Greska pri otvaranju datoteke");
13         exit(1);
14     }
15     for (int i=1;i<=10;i++)
16     {
17         fprintf(f,"%d\n",i*i);
18     }
19     fclose(f);
20     return 0;
21 }
```

	A
1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81
10	100

Formatirani način upisa i ispisa podataka

4. primjer: Pomoću formatiranog unosa učitati podatke iz datoteke kreirane u prošlom primjeru i ispisati ih na konzolu.

```
1  #include <iostream>
2  #include <cstdio>
3  #include <cstdlib>
4  using namespace std;
5
6  int main()
7  {
8      FILE*f;
9      int a;
10     f=fopen("kvadrati.xls","r");
11     if (f==NULL)
12     {
13         printf("Greska pri otvaranju datoteke");
14         exit(1);
15     }
16     while(!feof(f))
17     {
18         fscanf(f,"%d\n",&a);
19         printf("%d\n",a);
20     }
21     fclose(f);
22     return 0;
23 }
```

1
4
9
16
25
36
49
64
81
100

Ne stavimo li \n, onda nam dva puta čita zadnji redak

Naučili smo:

- upisati i čitati podatke uporabom formatiranog unosa i ispisa **scanf()** i **printf()**
- primijeniti **opise formata** kod formatiranog unosa i ispisa
- najaviti rad s datotekom pomoću **FILE***
- koristiti metode **fopen()** i **fclose()**
- upisati i čitati podatke iz datoteke korištenjem naredbi **fscanf()** i **fprintf()** uz primjenu prikladnih opisa formata

Datoteke



Izradila:
Danijela Ivanović – Ižaković
Elektrotehnička i prometna škola Osijek