

# Datoteke II

---

Tekstovne datoteke:

- Branje vrstic
- Branje znakov

Obdelava tekstovne datoteke

# Datoteke – branje po znakih

---

- Tekstovne datoteke lahko beremo tudi znak po znak - uporabljamo metodo *Read*. Ta nam vrne kodo znaka, ki ga preberemo. Ko bomo naleteli na konec datoteke, bo prebrana koda -1, ki jo uporabimo za to, da vemo, kdaj smo pri koncu. Seveda ga moramo pri izpisu pretvoriti v znak.

```
StreamReader s = File.OpenText(ime);
int beri;
beri = s.Read();
while (beri != -1) // konec datoteke
{
    Console.Write((char)beri); // izpisujemo ustrezne znake
    beri = s.Read();
}
s.Close();
```

# Zgled – Kopija datoteke po znakih

---

- ❑ Naredimo kopirajo datoteke tako, da beremo in zapisujemo znak po znak.

```
StreamReader beri = File.OpenText(vhod);
StreamWriter pisi = File.CreateText(izhod);
int prebranZnak;
prebranZnak = beri.Read();
while (prebranZnak != -1) // konec datoteke
{
    pisi.Write((char)prebranZnak); //
izpisujemo ustrezne znake
    prebranZnak = beri.Read();
}
beri.Close();
pisi.Close();
```

---

# Zgled: Kopija datoteke po vrsticah

---

- ❑ Ustvarimo kopijo datoteke tako, da beremo in zapisujemo cele vrstice

```
StreamReader beri = File.OpenText(vhod);
StreamWriter pisi = File.CreateText(izhod);
string vrst = beri.ReadLine();
while (vrst != null)
{
    Console.WriteLine(vrst);
    pisi.WriteLine(vrst);
    vrst = beri.ReadLine();
}
beri.Close();
pisi.Close();
```

---

# Zgled – primerjava dveh datotek

---

- Napišimo metodo *Primerjaj*, ki sprejme imeni dveh datotek in primerja njuno vsebino. Če sta vsebini datotek enaki, metoda vrne vrednost *true*, sicer vrne *false*. Predpostavimo, da sta imeni datotek ustrezni (torej, da datoteki za branje obstajata).

```
public static bool Primerjava(string ime1, string ime2)
{
    // odpremo obe datoteki za branje
    StreamReader dat1 = File.OpenText(ime1);
    StreamReader dat2 = File.OpenText(ime2);
    // preberemo vsebino prve in druge datoteke
    string beri1 = dat1.ReadToEnd();// v beri1 je vsebina prve
    datoteke
    string beri2 = dat2.ReadToEnd();// v beri2 je vsebina druge
    datoteke
    // zapremo oba tokova
    dat1.Close();
    dat2.Close();
    // primerjamo vsebini
    return (beri1.Equals(beri2));
}
```

---

# Vaje

---

- ❑ Dana je tekstovna datoteka DIJAKI.txt. V vsaki vrstici te datoteke je prvih 30 znakov rezervirano za ime dijaka, naslednjih 15 pa za učni uspeh ( od 1 do 5).
    - Koliko dijakov je v datoteki
    - Koliko dijakov ima splošni učni uspeh enak 5
    - Kolikšen je povprečen učni uspeh vseh dijakov
  
  - ❑ Dana je tekstovna datoteka Naloga.txt. V vsaki vrstici te datoteke so po tri cela števila, med seboj ločena s presledkom. Datoteko obdelaj tako, da za vsako vrstico na ekran izpišeš vsoto vseh treh števil, na koncu pa še skupno vsoto vseh števil!
-

# Vaje

---

- ❑ Napiši program, ki v poljubni tekstovni datoteki prešteje vse cifre (znake med 0 in 9) in na koncu izpiše, kolikokrat se vsaka cifra pojavi v tej datoteki. Ime datoteke programu podamo kot parameter ukazne vrstice.
  - ❑ Za poljubno tekstovno datoteko ugotovi in izpiši vse besede iz te datoteke in kolikokrat se posamezna beseda pojavi v tej datoteki. Pri tem ne delaj razlike med malimi in velikimi črkami!
  - ❑ Ustvarimo tekstovno datoteko *Kocka.txt*, v katero želimo shraniti rezultate 1000 metov kocke. V vsako vrstico te datoteke zapišimo zaporedno številko vrstice in naključno število med 1 in 6 (met kocke).
-

# Vaje

---

- Dana je tekstovna datoteka *Padavine.txt*. V njej je neznano število stavkov s podatki o količini padavin v določenem kraju. Vsaka vrstica je sestavljena iz imena kraja in količine letnih padavin. Med imenom kraja in količino padavin je ločilni znak '|'. Napišimo metodo za izpis vsebine datoteke na zaslon, metodo, ki dobi za parameter to datoteko in ki ugotovi ter izpiše skupno količino vseh padavin, ter metodo, ki vrne naziv kraja z največ padavinami. Na koncu napišimo še metodo *Dodaj*, ki na konec datoteke doda še dodatno vrstico: podatke o imenu kraja in količini padavin naj vnese uporabnik!
-



# Vaje:

---

- ❑ V datoteki *Manekenke.txt* so shranjeni podatki o velikosti manekenk. V vsaki vrsti je zapisana velikost manekenke v centimetrih (celo število), nato pa sledita ime in priimek. Polja so ločena z dvopičji. Primer datoteke:

179:Cindy:Crawford

182:Naomi:Campbel

185:Nina:Gazibara

180:Elle:Mac Perhson

180:Eva:Herzigova

---