

# Streamy pro soubory

Streamy pro soubory jsou objekty tříd `ifstream`, `ofstream`, `fstream`.

## Konstruktory

<code>ifstream()</code>
<code>ifstream(const char *filename, ios_base::openmode mode= ios_base::in)</code>
<code>ofstream()</code>
<code>ofstream(const char *filename, ios_base::openmode mode= ios_base::out)</code>
<code>fstream()</code>
<code>fstream(const char *filename, ios_base::openmode mode= ios_base::in   ios_base::out)</code>

Konstruktory bez parametrů pouze vytvoří objekt – neotevrou soubor.

Konstruktory s parametrem soubor otevrou (pokud přitom nenastane chyba).

Příznaky pro nastavení módu otevření (jsou v třídách `ios_base`, `ios`) :

<b>app</b>	( <b>append</b> ) Aktuální pozice pro zápis je nastavena na konec souboru.
<b>ate</b>	( <b>at end</b> ) Aktuální pozice je při otevření nastavena na konec souboru.
<b>binary</b>	( <b>binary</b> ) Do souboru budou ukládány binární údaje.
<b>in</b>	( <b>input</b> ) Ze souboru bude možné číst.
<b>out</b>	( <b>output</b> ) Do souboru bude možné zapisovat.
<b>trunc</b>	( <b>truncate</b> ) Současný obsah souboru je ignorován – délka souboru při otevření je 0.

## Funkce otevření a zavření

<code>is_open()</code>	Zjistí, zda soubor je otevřen.
<code>open(<i>parametry</i>)</code>	Otevře soubor. Argumenty funkce jsou stejné jako při otevření konstruktorem. Funkce nevrací žádnou hodnotu.
<code>close()</code>	Zavře soubor. Funkce nevrací žádnou hodnotu.

## Formátové čtení

<code>&gt;&gt;</code>	Formátové čtení
-----------------------	-----------------

## Neformátové čtení (vybrané funkce)

<code>gcount()</code>	Umožňuje zjistit, kolik bajtů bylo přečteno při poslední operaci neformátového čtení ( <i>get</i> , <i>getline</i> , <i>read</i> ...).
<code>get()</code>	Přečte znak a vrátí jeho hodnotu.
<code>get(char &amp;c)</code>	Přečte znak a uloží ho do proměnné <i>c</i> . Vrací referenci na stream.
<code>getline(char *s, streamsize n)</code>	Čte znaky a ukládá je do pole <i>s</i> . Čtení ukončí, jestliže <ul style="list-style-type: none"><li>bylo přečteno <i>n</i>-1 znaků</li><li>byl přečten znak '\n'</li><li>je konec souboru</li></ul> Je-li přečten znak '\n', je odstraněn ze vstupu, ale není přitom dán do pole <i>s</i> . Za přečtené znaky je v poli <i>s</i> vložena binární nula (výsledek tvoří řetězec). Vrací referenci na stream.
<code>getline(char *s, streamsize n, char delim)</code>	Stejně jako předchozí funkce. Místo znaku '\n' se čtení ukončí při přečtení znaku <i>delim</i> , který je odstraněn ze vstupu.
<code>peek()</code>	Zjistí, jaký je další znak na vstupu, a vrátí jeho hodnotu. Znak není přečten (není odstraněn ze vstupu).
<code>read(char *s, streamsize n)</code>	Čte znaky ze vstupu a ukládá je do pole <i>s</i> . Čtení se ukončí, jestliže <ul style="list-style-type: none"><li>bylo přečteno <i>n</i> znaků</li><li>je konec souboru</li></ul> Počet skutečně přečtených znaků lze zjistit funkcí <i>gcount</i> . Vrací referenci na stream.

## Formátový zápis

<<	Formátový zápis
----	-----------------

Pro formátový zápis lze používat všechny manipulátory již dříve uvedené u popisu streamu *cout*.

## Neformátový zápis (vybrané funkce)

<code>put(char c)</code>	Zapíše do streamu znak <i>c</i> . Vrací referenci na stream.
--------------------------	---

<code>write(const char *s, streamsize n)</code>	Zapíše blok <i>n</i> znaků. Vrací referenci na stream.
---	---

## Funkce pro zjištění stavu (vybrané funkce)

<code>good()</code>	Vrací <i>true</i> , je-li vše v pořádku.
<code>eof()</code>	Vrací <i>true</i> , jestliže v předchozí operaci byl dosažen konec souboru.
<code>fail()</code>	Vrací <i>true</i> , jestliže při některém čtení nastala chyba (jiná než konec souboru).
<code>!</code>	Operátor <code>!</code> má stejný význam jako funkce <b>fail</b> .
<code>bad()</code>	Vrací <i>true</i> , jestliže při některé operaci nastala závažná chyba.
<code>clear()</code>	Nuluje příznaky chyby.

### Příklad.

```
#include <fstream>
using namespace std;
try {
    { ofstream s("test", ios::binary);
      if (!s.is_open()) throw "Neotevrel se pro zapis.";
      int i=12;
      s.write((char *)&i, sizeof i);
      float f=3.2;
      s.write((char *)&f, sizeof f);
      char p[]="C++";
      s.write(p, sizeof p);
      if (!s.good()) throw "Chyba pri zapisu.";
      s.close();
    }
    { ifstream s("test", ios::binary);
      if (!s.is_open()) throw "Neotevrel se pro cteni.";
      int i;
      s.read((char *)&i, sizeof i);
      float f;
      s.read((char *)&f, sizeof f);
      char p[11];
      s.getline(p, 11, 0);
      if (s.eof()) throw "Konec souboru.";
      if (!s.good()) throw "Chyba pri cteni.";
      s.close();
      cout << i << " " << f << " " << p << endl;
    }
}
```

```
    }  
}  
catch (const char *s) { cout << s << endl; }
```

12 3.2 C++