Laboratorio di programmazione

Lezione VII

Tatiana Zolo

zolo@cs.unipr.it

INPUT E OUTPUT SU FILE

- `#include <fstream>`
- classe fstream:
  - ifstream: input file stream (per leggere).
  - ofstream: output file stream (per scrivere).

  Per aprire uno stream di input occorre dichiarare che appartiene alla classe ifstream: `ifstream in;`.

  Per aprire uno stream di output occorre dichiarare che appartiene alla classe ofstream: `ofstream out;`.

  Uno stream che eseguirà operazioni di input ed output deve appartenere alla classe fstream: `fstream entrambi;`.

IFSTREAM: APERTURA E CHIUSURA FILES

- nome del file:

```
char nome_file[] = ''prova.txt'';
```

  N.B. devono essere stringhe C (non C++)!

- aprire il file in lettura:

```
ifstream in_file(nome_file);
```

  oppure

```
ifstream in_file(''prova.txt'');
```

- controllo errori di apertura:

```
if (!in_file) { ... }
```

```
if (in_file.fail()) { ... }
```

- operazioni con l'oggetto istream definito.
- chiudere il file:

```
in_file.close();
```

IFSTREAM: LETTURA DATI

- Operatore >>:

```
in_file >> variabile_tipata;
```

  N.B. I caratteri di spaziatura vengono omessi!

- Funzione get(): dato c un carattere

  `c = in_file.get();` oppure `in_file.get(c);`

- Funzione (booleana) eof(): rilevamento della fine del file

```
while (!in_file.eof()) { ... }
```

  Restituisce “true” se è stata raggiunta la fine del file, cioè è stato letto l’ultimo carattere del file; restituisce “false” altrimenti.

  Carattere speciale EOF:

```
while ((c = in_file.get()) != EOF) ...;
```

es. `legge_file_get.cpp`, `legge_file_cin.cpp`.

OFSTREAM: APERTURA E CHIUSURA FILES

- nome del file:

```
char nome_file[] = ''prova.txt'';
```

  N.B. devono essere stringhe C (non C++)!

- aprire il file in scrittura: se non esiste lo crea, se esiste lo sovrascrive.
  `ofstream file_out(nome_file);` oppure
  `ofstream file_out(''prova.txt'');`.

  Modalità ios::app: l'output viene aggiunto alla fine del file a cui è destinato (non sovrascrive).

- controllo errori di apertura: `if (!out_file) { ... }` oppure
  `if (out_file.fail()) { ... }`.

- operazioni con l'oggetto ostream definito.

- chiudere il file:

```
out_file.close();
```

OFSTREAM: SCRITTURA DATI

- Operatore <<:

```
out_file << variabile_tipata;
```

- Funzione put():

```
out_file.put('a');
```

- Funzione eof(): è sempre falsa, si possono sempre aggiungere bytes.

es. `scrive_su_file.cpp`, `legge_file_cin.cpp`.

OPEN()

Si può definire un oggetto ifstream o ofstream senza specificare un file, che può essere associato in seguito mediante la funzione open().

```
ifstream in_file;
...
in_file.open(file_name);
// Controlla se open e' fallita.
if (!in_file) {
  ...
}
```

ESERCIZI

1. Copiare il contenuto di un file in un altro. Alla fine stampare a video il numero di bytes copiati. (`copia_file.cpp`.)
2. Riprendere l'esercizio n° 2 della lezione scorsa. Leggere da file 10 struct di persone con due campi, nome ed età, e restituire il nome della più giovane (o di una delle più giovani) con la relativa età (usare un'array di struct). Utilizzare una funzione che restituisca l'indice della persona più giovane presente nell'array. (`piu_giovane_file.cpp`.)
3. Esercizio tipo esame:

   `http://www.math.unipr.it/~gianfr/Teaching/FondProgr/compiti.html`

   Compito del 23/6/2004.