

C++

التعامل مع الملفات

معالجة الملفات <

كلاسات الحزمة `fstream` <

طريقة فتح و إغلاق ملف <

التشبيك على حالة الكائن الذي نتعامل من خلاله مع الملف <

أمثلة شاملة على التعامل مع الملفات <

organized by
walid.info

معالجة الملفات

التعامل مع الملفات أو معالجة الملفات (**Files Handling**) يقصد منها إجراء عملية ما على الملفات الموجودة في حاسوب المستخدم كقراءة محتوى ملف و عرضه في البرنامج، إنشاء نسخة منه، تعديل محتواه أو حذفه، سواء كان

نوع الملف `txt` ، `jpg` ، `mp4` أو أي نوع آخر.

الآن، للتعامل مع الملفات يجب تضمين الحزمة `<fstream>` لأنها تحتوي على الكلاسات المخصصة لذلك، بالإضافة

للحزمة `<iostream>` لأننا سنحتاج منها العامل `<<` عند الكتابة في الملف.

إذاً يجب كتابة هذين السطرين عند التعامل مع الملفات.

```
#include <iostream>
#include <fstream>
```



مصطلحات تقنية

- إسم الحزمة `<fstream>` مشتق من جملة **File Stream** و التي تعني أنها مخصصة للتعامل مع الملفات.
- إسم الحزمة `<iostream>` مشتق من جملة **Input Output Stream** و التي تعني أنها تحتوي على أوامر الإدخال و الإخراج سواء على الشاشة أو في الملفات.

كلاسات الحزمة `fstream`

الحزمة `<fstream>` تحتوي على الكلاسات الأساسية التالية التي يمكن استخدامها للتعامل مع الملفات.

الكلاس مع تعريفه	
<code>ofstream</code>	1 يستخدم لإنشاء كائن يتيح لنا إمكانية إنشاء ملف جديد و الكتابة فيه.
<code>ifstream</code>	2 يستخدم لإنشاء كائن يتيح لنا إمكانية قراءة محتوى الملف.
<code>fstream</code>	3 يستخدم لإنشاء كائن يتيح لنا إمكانية إنشاء ملف جديد, الكتابة فيه و القراءة منه أيضاً. إذاً هذا الكلاس يعتبر دمج للكلاس <code>ofstream</code> و الكلاس <code>ifstream</code> .

طريقة فتح و إغلاق ملف

إذا أردت قراءة محتوى ملف أو الكتابة فيه فلا بد من أن يكون مفتوحاً من قبل برنامجك نفسه حتى تتمكن من ذلك.

الكلاسات الثلاثة `ifstream` و `ofstream` و `fstream` جميعها تحتوي على دالة إسمها `open()` نستخدمها

لنفتح الملف الذي نريد التعامل معه.

3

بناء الدالة `open()`

```
void open(const char *filename, ios::openmode mode)
```



- مكان الباراميتر `filename` نمرر إسم و مسار الملف الذي نريد فتحه كنص عادي.
- هو باراميتر اختياري يمكننا أن نمرر مكانه ثابت أو أكثر من الثوابت الجاهزة في الكلاس `ios` حتى نحدد للمترجم لماذا نريد فتح الملف.

في الجدول التالي وضعنا أسماء ثوابت الكلاس `ios` التي يمكنك تمريرها مكان البارميتر `mode`.

الثابت مع تعريفه

<code>ios::app</code>	1	يستخدم لإعلام المترجم بأن المحتوى الجديد الذي سيتم إضافته سيوضع في آخر الملف.
<code>ios::ate</code>	2	يستخدم لإعلام المترجم بأن سيتم فتح الملف بهدف الكتابة و القراءة منه مع الإشارة إلى أنه سيبدأ من آخره.
<code>ios::in</code>	3	يستخدم لإعلام المترجم بأنه سيتم فتح الملف بهدف القراءة منه.
<code>ios::out</code>	4	يستخدم لإعلام المترجم بأنه سيتم فتح الملف بهدف الكتابة فيه.

`ios::trunc`

يستخدم لإعلام المترجم بأنه في حال كان الملف موجود مسبقاً، سيتم مسح محتواه عند فتحه.

عند استدعاء الدالة `open()` يمكنك استخدام العامل `|` في حال أردت أن تمرر لها أكثر من قيمة مكان البارميتر

mode كالتالي.

```
ofstream myfile;
myfile.open ("example.txt", ios::out | ios::app);
```

o
ro
g
a
n
i
z
e
d
 b
y
w
a
l
i
d
.i
n
f
o

معلومة تقنية

- الكلاس `ofstream` يستخدم الثابت `ios::out` بشكل افتراضي.
- الكلاس `ifstream` يستخدم الثابت `ios::in` بشكل افتراضي.
- الكلاس `fstream` لا يستخدم أي ثابت بشكل افتراضي.

أهمية إغلاق الملف عند الإنتهاء منه

عند الإنتهاء من التعامل مع أي ملف، قم بإغلاقه على الفور لأن ذلك من شأنه تحسين أداء البرنامج حيث سيخفف من حجم المساحة المحجوزة للملف في الذاكرة بالإضافة إلى أنك تصبح قادر على التعامل مع هذا الملف بشكل مباشر من خارج برنامجك.

الكلاسات الثلاثة `ifstream` و `ofstream` و `fstream` جميعها تحتوي على دالة إسمها `close()` نستخدمها لإغلاق الملف.

إذا لإغلاق الإتصال مع أي ملف مفتوح, يجب أن تستدعي الدالة `close()` من الكائن الذي بالأساس فتحت الملف من

5

خلاله.

التشبيك على حالة الكائن الذي تتعامل من خلاله مع الملف

عند استخدام الدالة `open()` لفتح الملف سواء بهدف القراءة أو الكتابة فيه فإن ذلك قد لا ينجح دائماً.

فمثلاً إذا كنت تريد إنشاء ملف جديد, قد تكون لا تملك صلاحية لإنشاء ملف في الحاسوب أو لا يوجد مساحة كافية لإنشاء الملف فيها, أو أن الملف موجود أصلاً و لكنه مفتوح من قبل برنامج آخر. و في حال كنت تريد قراءة محتوى ملف موجود في الحاسوب قد تواجه أيضاً بعض المشاكل, كأن يكون مسار الملف الموضوع غير صحيح, أو أن الملف مفتوح من قبل برنامج آخر أو أنك لا تملك صلاحية للقراءة منه إلخ..

بعد إنشاء الكائن الذي ستتعامل من خلاله مع الملف, يمكنك استخدام الجمل الشرطية `if` و `else` بكل سهولة كالتالي لمعرفة ما إن كان يمكنك التعامل مع الملف أم لا.

```
ofstream myfile;
myfile.open ("example.txt");

if (myfile)
{
    // إذا كان الإتصال بالملف لا يوجد فيه مشاكل, سيتم تنفيذ الأوامر التي نضعها هنا
}
else
{
    // إذا أردت إعلام المستخدم بأنه حدث مشكلة أثناء الإتصال بالملف, فيمكنك كتابة ذلك هنا
}
```

هناك 4 دوال جاهزة يمكنك استخدامها للتأكد من أن الإتصال بالملف سليم و أنه لم تحدث أي مشكلة عند التعامل معه سواء عند القراءة أو الكتابة فيه.

bool bad()

تستخدم لمعرفة ما إن حصلت أي مشكلة عند القراءة أو الكتابة في الملف.
ترجع **true** إذا حدثت مشكلة و ترجع **false** إذا لم تحدث أي مشكلة.

1

من المشاكل التي نقصدها كأن تحاول الكتابة في الملف و لكنه لا يوجد مساحة كافية للتخزين، أو في حال فتحت ملف بواسطة كائن من `ofstream` و لكنك كنت تنوي استخدامه للقراءة و ليس للكتابة. في هذه الحالات يمكنك الإستفادة من هذه الدالة لمعرفة ما إن حدث خطأ أم لا.

bool fail()

مثل الدالة `bad()` تماماً بالإضافة إلى أنها تشيك على المشاكل التي قد تحدث عند التعامل مع محتوى الملف.
على سبيل المثال، إذا قمت بقراءة عدد مخزن في الملف و قمت بقراءته في برنامجك و من ثم التعامل معه كأنه عدد عادي بدون أن تحوله لعدد ستجد أنها تنبهك عن هذا الخطأ أيضاً.
ترجع **true** إذا حدثت مشكلة و ترجع **false** إذا لم تحدث أي مشكلة.

2

bool eof()

إسم الدالة هو اختصار لجملة **End Of File** و هي تستخدم لمعرفة ما إن وصلت في القراءة أو الكتابة إلى آخر الملف أم لا.
ترجع **true** إذا كان المترجم وصل لنهاية الملف و ترجع **false** إذا يصل بعد لنهايته.

3

bool good()

تستخدم لمعرفة ما إن حصلت أي مشكلة كانت عند التعامل مع الملف و هي تشمل كل أنواع المشاكل التي قد تحدث.
ترجع **true** إذا لم تحدث أي مشكلة و ترجع **false** إذا حدثت مشكلة ما.

4

الكلاسات الثلاثة `ifstream` و `ofstream` و `fstream` جميعها تحتوي على هذه الدوال و ستجد كيفية استخدامها لاحقاً في الأمثلة.



organized by
walid.info

