

Klasse pfc::bitmap

Kurzbeschreibung für die Version 2.1.0

Peter Kulczycki

peter.kulczycki@fh-hagenberg.at

Department of Software Engineering
University of Applied Sciences Upper Austria
Softwarepark 11, 4232 Hagenberg, Austria

Version 1.10.1234 – 2. Oktober 2016

- 1 Interface
- 2 Beispiel
- 3 Anmerkungen und Einschränkungen
- 4 Konzeptbilder

Interface / Konstruktoren

```
1  bitmap ()
2  bitmap (bitmap const & src)
3  bitmap (bitmap && src)
4
5  bitmap (int const width, int const height)
6  bitmap (int const width, int const height,
7          pfc::byte_t * const p_image)
8
9  bitmap (char const * const p_filename)
10 bitmap (string const & filename)
11
12 bitmap (istream & in)
```

Der Default-Konstruktor erzeugt ein Bitmap (Datentyp `pfc::bitmap`) der Größe 0×0 Pixel.

Mit den Methoden `create`, `from_file` oder `from_stream` bzw. dem Zuweisungsoperator kann dessen Größe verändert werden. Die Parameter `width` und `height` geben die Größe (in Pixel) des zu konstruierenden Bitmaps an. Das mit `width` und `height` konstruierte Bitmap ist weiß.

Mit `p_image` kann ein extern allozierter Speicher für die Bilddaten vorgegeben werden.

Interface / Initialisierung

```
1 void clear ()
2 void create (int const width, int const height)
3 void create (int const width, int const height,
4             pfc::byte_t * p_image)
```

Die Methode `clear` ist ein Synonym für den Methodenaufruf `create(0,0)`.

Mit der Methode `create` kann die Größe eines Bitmaps verändert werden. Die Parameter `width` und `height` geben die neue Größe eines Bitmaps (in Pixel) an. Das so veränderte Bitmap ist weiß. (Auch dann, wenn die neuen Größen gleich den alten sind.)

Mit `p_image` kann ein extern allozierter Speicher für die Bilddaten vorgegeben werden.

Interface / Operatoren und Getter

```
1  bitmap & operator = (bitmap const & rhs)
2  bitmap & operator = (bitmap && rhs)
```

```
1  int get_height () const
2  int get_width () const
3
4  int get_size () const
5  int get_stride () const
6
7  pfc::byte_t *      get_image ()
8  pfc::byte_t const * get_image () const
9
10 pixel_t *      get_pixels ()
11 pixel_t const * get_pixels () const
```

Die Methode `get_size` liefert die Anzahl der Bytes, die die geladenen Bilddaten im Speicher benötigen.

Die Methode `get_stride` liefert die Anzahl der Bytes, aus denen eine Bildzeile in der `bmp`-Datei besteht.

Die Methoden `get_image` bzw. `get_pixels` liefern Zeiger auf die Bilddaten.

Interface / Lesen und schreiben

```
1  bool from_file    (char const * const p_filename)  
2  bool from_file    (string const & filename)  
3  bool from_stream  (istream & in)
```

```
1  bool to_file      (char const * const p_filename) const  
2  bool to_file      (string const & filename) const  
3  bool to_stream    (ostream & out) const
```

Diese Methoden lesen bzw. schreiben ein Bitmap. Es können dabei Dateinamen (Parameter `p_filename` und `filename`) oder Dateiströme (Parameter `in` und `out`) angegeben werden. Dateinamen müssen die Erweiterung „bmp“ besitzen.

Alle Funktionen liefern dann `true`, wenn die Lese- bzw. Schreiboperationen auf Stream-Ebene erfolgreich durchgeführt wurden.

Beispiel

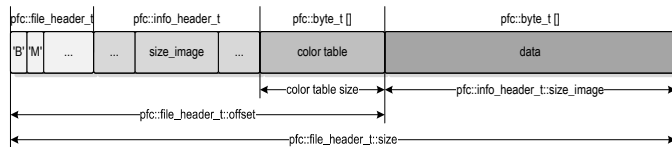
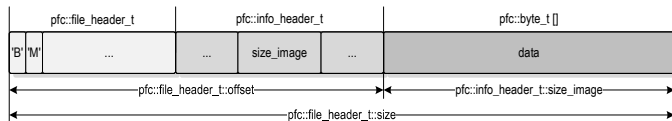
```
1  #include "pfc-bitmap.h"
2
3  int main () {
4      pfc::bitmap b1 (300, 100);
5      pfc::bitmap b2;
6      pfc::bitmap b3;
7
8      b2 = b1;
9
10     b3.from_file ("./erythrocytes.bmp");
11
12     pfc::swap (b2, b3);
13
14     b1.to_file ("./output-1.bmp");    // <white>
15     b2.to_file ("./output-2.bmp");    // erythrocytes.bmp
16     b3.to_file ("./output-3.bmp");    // <empty>
17 }
```

Anmerkungen und Einschränkungen

Für die Version 2.1.0 gelten die folgenden Einschränkungen:

- 1 Die Klasse `pfc::bitmap` ist mit MS Visual-Studio 2013 bzw. 2015 übersetzbar.
- 2 Es werden nur 24-Bit-Bitmaps unterstützt (lesend und schreibend).
- 3 Die dreifache Breite eines Bitmaps (in Pixel) muss durch vier teilbar sein (eine Scanline muss an einer 32-Bit-Grenze enden).

Konzeptbilder



Konzeptbilder

