

# Das Bild sichern

Wie sichert man  
den Inhalt des Grafikfensters?

# Das Bild sichern

- Die Klasse Zeichenflaeche stellt eine Methode bereit, um das dargestellte Bild als Bilddatei abspeichern zu können.

```
def BildExport(self, dateiname):
    '''exportiert im png- oder jpg-Format'''
    self.InitBuffer()
    typ=dateiname.partition('.')[2].upper()
    if not(typ in ('PNG','JPG')): return 'Dateityp png oder jpg'
    self.__image=self._buffer.ConvertToImage()
    bitmap_type={'PNG':wx.BITMAP_TYPE_PNG,
                 'JPG':wx.BITMAP_TYPE_JPEG}[typ]
    try:
        with open(dateiname, 'wb') as bild_datei:
            self.__image.SaveFile(dateiname,bitmap_type)
    except IOError as ioerr:
        return 'Dateifehler: ' + str(ioerr)
```

# Das Bild sichern

- Zulässige Formate der Bilddatei sind png oder jpg, dazu wird der übergebene Dateiname überprüft.

```
def BildExport(self, dateiname):
    '''exportiert im png- oder jpg-Format'''
    self.InitBuffer()
    typ=dateiname.partition('.')[2].upper()
    if not(typ in ('PNG','JPG')): return 'Dateityp png oder jpg'
    self.__image=self._buffer.ConvertToImage()
    bitmap_type={'PNG':wx.BITMAP_TYPE_PNG,
                 'JPG':wx.BITMAP_TYPE_JPEG}[typ]
    try:
        with open(dateiname, 'wb') as bild_datei:
            self.__image.SaveFile(dateiname,bitmap_type)
    except IOError as ioerr:
        return 'Dateifehler: ' + str(ioerr)
```

# Das Bild sichern

- Die Methode `ConvertToImage()` erzeugt ein image – Objekt (*bitmap*, keine Vektorgrafik).

```
def BildExport(self, dateiname):
    '''exportiert im png- oder jpg-Format'''
    self.InitBuffer()
    typ=dateiname.partition('.')[2].upper()
    if not(typ in ('PNG','JPG')): return 'Dateityp png oder jpg'
    self.__image=self._buffer.ConvertToImage()
    bitmap_type={'PNG':wx.BITMAP_TYPE_PNG,
                 'JPG':wx.BITMAP_TYPE_JPEG}[typ]
    try:
        with open(dateiname, 'wb') as bild_datei:
            self.__image.SaveFile(dateiname,bitmap_type)
    except IOError as ioerr:
        return 'Dateifehler: ' + str(ioerr)
```

# Das Bild sichern

- Da beim Schreiben von Daten in Dateien Fehler auftreten können, muss mit try – except gekapselt werden.

```
def BildExport(self, dateiname):
    '''exportiert im png- oder jpg-Format'''
    self.InitBuffer()
    typ=dateiname.partition('.')[2].upper()
    if not(typ in ('PNG','JPG')): return 'Dateityp png oder jpg'
    self.__image=self._buffer.ConvertToImage()
    bitmap_type={'PNG':wx.BITMAP_TYPE_PNG,
                 'JPG':wx.BITMAP_TYPE_JPEG}[typ]

    try:
        with open(dateiname, 'wb') as bild_datei:
            self.__image.SaveFile(dateiname,bitmap_type)
    except IOError as ioerr:
        return 'Dateifehler: ' + str(ioerr)
```

# Das Bild sichern

- Das eigentliche Schreiben übernimmt die Methode SaveFile des image-Objekts.

```
def BildExport(self, dateiname):
    '''exportiert im png- oder jpg-Format'''
    self.InitBuffer()
    typ=dateiname.partition('.')[2].upper()
    if not(typ in ('PNG','JPG')): return 'Dateityp png oder jpg'
    self.__image=self._buffer.ConvertToImage()
    bitmap_type={'PNG':wx.BITMAP_TYPE_PNG,
                 'JPG':wx.BITMAP_TYPE_JPEG}[typ]
    try:
        with open(dateiname, 'wb') as bild_datei:
            self.__image.SaveFile(dateiname,bitmap_type)
    except IOError as ioerr:
        return 'Dateifehler: ' + str(ioerr)
```