Praktikum Objektorientierte Programmierung in C++ (WS 2021/2022)

A4 Teil 1: Hausaufgabe zur Vorbereitung auf die Präsenz-Gruppe/Part 1: Homework Task for Preparation of the Presence Group

Am Beispiel von Grafiken im ASCII-kodierten Portable-Pix-Map-Format (Datei-Endung `.ppm`, siehe auch die separate Erläuterung dazu) soll in dieser Aufgabe eine Klasse definiert werden, über die ein Pixel-Rasterbild im RGB-Farbmodell gespeichert und bearbeitet werden kann. In der Klasse enthalten sind Member-Funktionen zum Lesen und Schreiben einer `.ppm`-Text-Format-Bilddatei sowie eine Funktion zur Bildbearbeitung als Beispiel. Programmieren Sie im einzelnen:/
Using the example of graphics in ASCII-coded portable pix map format (file extension `.ppm`, see also the separate explanation), in this task a class shall be defined storing a pixel raster image in the RGB colour model and also be edited. The class has member functions for reading and writing a `.ppm` text format picture file as well as a function for image processing as an example. Program in detail:

1. Definieren Sie eine C++-Struktur mit Namen **`Pixel`** mit drei vorzeichenlosen ganzen Zahlen für den Rot-, Grün- und Blau-Anteil eines Grafik-Punktes./
   Define a C++ structure called **`Pixel`** with three unsigned integer values for the red, green and blue component of one graphic point.
2. Definieren Sie eine Klasse mit Namen **`Image`** mit folgenden Membern:/
   Define a class with name **`Image`** with following members:
   - zwei private Zeichen-Attribute für die ersten beiden sogenannten magischen Zahlen einer `.ppm`-Datei./
     two private character attributes for the first to so called magic numbers of a `.ppm`-file.
   - drei private vorzeichenlose ganzen Zahlen für die Breite, Höhe und den jeweils maximalen Farbwert./
     three private unsigned integers for the width, height and the maximum colour value.
   - privater Zeiger auf einen Zeiger vom Typ Pixel (also **`Pixel **r`**; dieser soll im Lauf des Programms auf das Pixelraster zeigen gespeichert als ein dynamisch erzeugtes quasi zweidimensionales Feld von Bildpunkten)./
     private pointer to a pointer of type Pixel (i.e. **`Pixel **r`**; during program runtime it will point to the pixel raster image stored as a dynamically generated quasi two-dimensional array of pixels).
   - öffentlicher Standard-Konstruktor, in dem die beiden Magic-Number-Zeichen auf **`P`** und **`3`** initialisiert werden, die Breite und Höhe auf **`0`**, der maximale Farbwert auf **`255`** und der Zeiger auf einen Nullzeiger./
     public standard constructor, in which the two magic number characters are initialised to **`P`** and **`3`**, the width and height to **`0`**, the maximum colour value to **`255`** and the pointer to a null pointer.
   - private Member-Funktion mit Namen **`alloc_raster`** ohne Parameter und ohne Rückgabe. Im Rumpf soll Speicherplatz auf dem Heap für ein quasi zweidimensionales Feld von **`Pixel`**n in der durch die Attribute für die Breite und Höhe gegebenen Größe reserviert werden; dessen Adresse soll dem Zeiger zugewiesen werden (siehe Beispiel-Kode-Fragment unten). Überprüfen Sie zu Beginn zusätzlich, dass der Zeiger ein Nullzeiger ist und die Werte für die Breite und Höhe größer als Null sind./
     private member function named **`alloc_raster`** with no parameters and no return. In the body, memory space on the heap shall be reserved for a quasi two-dimensional array of **`Pixel`**s in the size given by the attributes for the width and height; its address shall be assigned to the pointer (see example code fragment below). In addition, check before that the pointer is a null pointer and the values for the width and height are greater than zero.
   - öffentliche Member-Funktion mit Namen **`read_ppm`** mit einer C++-Zeichenkette für den Dateinamen als Parameter und ohne Rückgabe. Im Rumpf soll die entsprechende ASCII-kodierte `.ppm`-Textdatei geöffnet, der Header der Datei eingelesen und dessen Werte in den Attributen gespeichert werden. Dann soll ein Feld von Pixeln in der eingelesenen Größe dynamisch allokiert werden, in das alle Pixel mit ihrem Rot-, Grün- und Blau-Wert der Reihe nach aus der Datei eingelesen und gespeichert und die Datei wieder geschlossen werden./
     public member function named **`read_ppm`** with a C++ string for the file name as parameter and without return. In the body, the corresponding ASCII-encoded `.ppm` text file shall be opened, the header of the file shall be read in and the values be stored in the attributes. Then an array of pixels of the size read in shall be dynamically allocated, row by row all pixels with their red, green and blue values shall be read from the file and stored into it and the file be closed.
   - öffentliche Member-Funktion mit Namen **`write_ppm`** mit einer C++-Zeichenkette für den Dateinamen als Parameter und ohne Rückgabe. Im Rumpf soll die entsprechende ASCII-kodierte `.ppm`-Textdatei schreibend geöffnet, der Header der Datei gefolgt von einer Kommentarzeile **`"# end of header"`** in die Datei geschrieben werden, zeilenweise die Rot-, Grün- und Blau-Werte der Pixel im

Feld und die Datei geschlossen werden./
public member function named **`write_ppm`** with a C++ string for the file name as parameter and without return. In the body, the corresponding ASCII-encoded **`.ppm`** text file shall be opened in write mode, the header of the file followed by a comment line **`"# end of header"`** be written into the file, then the red, green and blue values of the pixels in the array shall be written line by line and the file be closed.

- öffentliche Member-Funktion mit Namen **`mirror_vertically`** ohne Parameter und ohne Rückgabe. Im Rumpf soll das Bild vertikal gespiegelt werden, also in jeder Zeile das erste Pixel mit dem letzten, das zweite mit dem vorletzten, usw. vertauscht werden./
  public member function named **`mirror_vertically`** without parameter and without return. In the body, the image shall be mirrowed vertically, i.e. in each row the first with the last pixel, the second with the second to last, etc. be exchanged.

3. Schreiben Sie eine **`main`**-Funktion mit folgenden Definitionen und Anweisungen:/
   Write a **`main`** function with following definitions and statements:
   - definieren Sie eine Objekt-Variable vom Typ **`Image`** und alle weiteren Variablen, die sie benötigen./
     define an object variable of type **`Image`** and all further variables you need.
   - implementieren Sie ein kleines Menü mit den unten in den Beispielen gezeigten vier Funktionalitäten/
     implement a small menu with the four functionalities shown below in the examples
     - zum Beenden des Programms,/
       for exiting the program,
     - zum Lesen und zum Schreiben einer **`.ppm`**-Textdatei mit einem Bild, wo jeweils ein Dateiname eingegeben und eine entsprechende Nachricht an das Bild-Objekt geschickt werden soll,/
       reading and writing a **`.ppm`** text file, where a file name shall be inputted and an appropriate message shall be send to the picture object,
     - dem vertikalen Spiegeln des Bilds über eine Nachricht an das Bild-Objekt./
       vertically mirroring the image by sending a message to the picture object.

Hinweis 1/Hint 1

Member-Funktion zur dynamischen Speicherplatz-Allokation des quasi zweidimensionales Felds von Pixeln im Heap./
Member function for dynamic memory allocation of the quasi two-dimensional array of Pixels on the heap

```
void alloc_raster()
{  // ...
   r = new Pixel*[height]; // allocate pointers to the rows
   for (unsigned int k = 0; k < height; k++) // for all rows
      r[k] = new Pixel[width]; // allocate pixels in row
   // ...
 }
```

Hinweis 2/Hint 2

Während der Entwicklung Ihres Programms kann es sehr hilfreich sein, Ausgaben der gelesenen oder der geschriebenen Header-Daten und ersten Bildpunkte zusätzlich auf dem Bildschirm auszugeben sowie sich die **`.ppm`**-Dateien auch mit einem ASCII-Texteditor anzuschauen. Schauen Sie sich auch die von Ihnen geschriebenen **`.ppm`**-Dateien mit einem Bildbetrachtungsprogramm wie bspw. IrfanView an./

During the development of your program, it can be very helpful to additionally output the read or the written header data and first pixel data onto the screen as well as to also view the **`.ppm`** files with an ASCII text editor.

Look at your written **`.ppm`** files with an image viewer such as e.g. IrfanView.

Beispiel Programmlauf 1/Example Program Run 1

```
0 Ende                / end
1 lese .ppm Datei     / read .ppm file
2 vertikal spiegeln   / mirror vertically
3 schreibe .ppm Datei / write .ppm file
>> 1
please input file name: campus_duisburg.ppm
read_ppm:  campus_duisburg.ppm opened...
           read header of file...
           allocate 220x147 pixel array on heap... done
           read RGB pixel data row by row...
           done and campus_duisburg.ppm closed

0 Ende                / end
1 lese .ppm Datei     / read .ppm file
2 vertikal spiegeln   / mirror vertically
3 schreibe .ppm Datei / write .ppm file
>> 2
           start mirror image vertically...
           done

0 Ende                / end
1 lese .ppm Datei     / read .ppm file
2 vertikal spiegeln   / mirror vertically
3 schreibe .ppm Datei / write .ppm file
>> 3
please input file name: campus_duisburg_mirrowed.ppm
write_ppm: campus_duisburg_mirrowed.ppm opened...
           write header of file...
           write RGB pixel data row by row...
           done and campus_duisburg_mirrowed.ppm closed

0 Ende                / end
1 lese .ppm Datei     / read .ppm file
2 vertikal spiegeln   / mirror vertically
3 schreibe .ppm Datei / write .ppm file
>> 0
```

Beispiel Programmlauf 2/Example Program Run 2

```
0 Ende                / end
1 lese .ppm Datei     / read .ppm file
2 vertikal spiegeln   / mirror vertically
3 schreibe .ppm Datei / write .ppm file
>> 1
please input file name: peppers.ppm
read_ppm:  peppers.ppm opened...
           read header of file...
           allocate 512x384 pixel array on heap... done
           read RGB pixel data row by row...
           done and peppers.ppm closed

0 Ende                / end
1 lese .ppm Datei     / read .ppm file
2 vertikal spiegeln   / mirror vertically
3 schreibe .ppm Datei / write .ppm file
>> 2
           start mirror image vertically...
           done

0 Ende                / end
1 lese .ppm Datei     / read .ppm file
2 vertikal spiegeln   / mirror vertically
3 schreibe .ppm Datei / write .ppm file
>> 3
please input file name: peppers_mirrowed.ppm
write_ppm: peppers_mirrowed.ppm opened...
           write header of file...
           write RGB pixel data row by row...
           done and peppers_mirrowed.ppm closed

0 Ende                / end
1 lese .ppm Datei     / read .ppm file
2 vertikal spiegeln   / mirror vertically
3 schreibe .ppm Datei / write .ppm file
>> 0
```

Last modified: Friday, 26 November 2021, 12:09 AM

◄ RGB-Farbmodell und Bild-Dateien

Jump to...

campus_duisburg.ppm ►