

Ein- und Ausgabe in C++

Lernziele

Die in der Vorlesung behandelten Klassen `string`, `iostream`, `omanip` und `fstream` aus der Standard-Bibliothek sollen vertieft und angewendet werden.

Aufgabe 4:

Implementieren Sie den Heap aus Aufgabe 2 als Klasse in C++.

- Ersetzen Sie die Funktion `createHeap` durch einen Konstruktor und die Funktion `destroyHeap` durch einen Destruktor.
- Realisieren Sie die Speicherverwaltung mit `new` und `delete`.

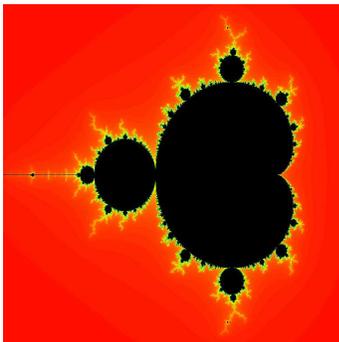
Ergänzen Sie die Klasse `Heap` um eine Methode

- `readFile(std::string filename)`, die die angegebene Datei öffnet, die darin enthaltenen Werte einliest und in den Heap übernimmt.
- `writeFile(std::string filename)`, die die angegebene Datei öffnet und alle Werte des Heaps darin speichert.

Wie immer, gehört das Erstellen von Testtreibern und Testfällen zur Aufgabe, um die Korrektheit Ihrer Implementierung nachzuweisen.

Zusatz-Aufgabe Z4:

Komplexe Zahlen eignen sich nicht nur für die komplexe Wechselstromrechnung, wir nutzen sie auch zur Darstellung der Mandelbrot-Menge¹, die im allgemeinen Sprachgebrauch oft auch Apfelmännchen genannt wird.



Die Mandelbrot-Menge M ist die Menge aller komplexen Zahlen c , für welche die rekursiv definierte Folge komplexer Zahlen z_0, z_1, z_2, \dots mit dem Bildungsgesetz

$$z_{n+1} = z_n^2 + c$$

und dem Anfangsglied $z_0 = 0$ beschränkt bleibt, das heißt, der Betrag der Folgenglieder wächst nicht über alle Grenzen.

¹<https://de.wikipedia.org/wiki/Mandelbrot-Menge>

Die grafische Darstellung dieser Menge erfolgt in der komplexen Ebene. Die Punkte der Menge \mathbb{M} werden dabei in der Regel schwarz dargestellt und der Rest farbig, wobei die Farbe eines Punktes den Grad der Divergenz der zugehörigen Folge widerspiegelt. Man berechnet also den Wert n , für den der Betrag $|z_n|$ z.B. größer als zwei ist, und diesem Wert n ordnet man eine Farbe zu.

Schreiben Sie ein Programm, das die Farbwerte in eine Datei schreibt, sodass man die Apfelmännchen mittels eines Bildbetrachtungsprogramms ansehen kann. Zur einfachen Speicherung von Rastergrafiken eignen sich das Portable-Pixmap- (PPM²) und das Windows-Bitmap-Format (BMP³).

Sinnvoll ist es, dem Programm fünf Parameter zu übergeben:

- Die maximale Anzahl von Iterationsschritten. Wird dieser Wert überschritten, so wird das entsprechende Pixel der Menge \mathbb{M} zugeordnet. Der Wert sollte mindestens 100 betragen. Bei stärkerer Vergrößerung sind zur korrekten Darstellung der Strukturen teilweise erheblich größere Werte erforderlich und es ergeben sich dann deutlich längere Rechenzeiten.
- Die beiden Werte re_{min} und im_{min} beschreiben den unteren, linken Bildpunkt, die beiden Werte re_{max} und im_{max} beschreiben den oberen, rechten Bildpunkt des Apfelmännchens. Die Berechnung des dem Pixel zugeordneten komplexen Zahlenwerts $c = (c_{re} + i \cdot c_{im})$ mit dem Realteil $re_{min} \leq c_{re} \leq re_{max}$ und dem Imaginärteil $im_{min} \leq c_{im} \leq im_{max}$ erfolgt durch lineare Interpolation zwischen den beiden Punkten.

Geeignete Programm-Parameter sind:

<code>iter</code>	128	256	384	512	640	768
<code>r_min</code>	-2.0	-1.0	-0.78	-0.751	-0.7446	-0.74401
<code>i_min</code>	-1.2	0.0	0.10	0.118	0.1303	0.13151
<code>r_max</code>	0.8	-0.5	-0.74	-0.735	-0.7424	-0.74327
<code>i_max</code>	1.2	0.5	0.14	0.134	0.1325	0.13225

²https://de.wikipedia.org/wiki/Portable_Anymap

³https://de.wikipedia.org/wiki/Windows_Bitmap