

A1 BS Theorie

Martin Behm

May 6, 2019

0.1 Was ist ein CGA-Bildschirm?

Color Graphics Adapter oder kurz CGA ist ein Grafikkartenstandard aus frühen PC-Zeiten.

0.2 Wie viele Zeichen passen auf einen CGA-Bildschirm und in welcher Anordnung (Zeilen und Spalten)?

Es passen insgesamt $80 \cdot 25 = 2000$ Zeichen mit jeweils 2 Byte, also $2000 \cdot 2 = 4000$ auf den Bildschirm, wobei 80 Spalten mit 25 Zeilen vorhanden sind.

0.3 Wie viel Platz (in Byte) benötigt ein Zeichen auf dem CGA-Bildschirm (inklusive aller Attribute)?

2 Byte. Das sind 16 Bits. Der Gerade Offset nimmt den ASCII-Code des darzustellenden Zeichens auf, während die ungeraden Offsets für die Vordergrund- und Hintergrundfarbe benötigt werden.

0.4 Wie kann man den CGA-Bildschirm ansprechen?

Man greift mithilfe der Adresse **0xb8000** auf den Inhalt des CGAs zurück, also dem **Offset** ≥ 0 .

0.5 Wie sieht der Ablauf des Schreibens eines Zeichens skizzenhaft aus?

Zuerst muss die Vordergrundfarbe (*die ersten 4 Bits (0 – 3)*), dann die Hintergrundfarbe (*die nächsten 3 Bits (4 – 6)*) und ob der Character blinken soll (*Bit 7*), bestimmt werden.

Beispiel $x = 1.100.1000$ würde bedeuten, dass der Char x eine *rote* Hintergrundfarbe und eine *weiße* Vordergrundfarbe mit zusätzlichem Blinken hat.

0.6 Wie sieht der Ablauf des Setzens des Cursors skizzenhaft aus?

Geben mithilfe des Indexregisters den Index des gewünschten Steuerregister aus (*14* steht für **Low Offset**, *15* für **High Offset**, also für große Adressen), um dann auf das so adressierte Steuerregister zugreifen zu können. (*Ports sind in Hex*)

0.7 Was ist eine Initialisierungsliste in C₊₊?

Sie regelt die Konstruktion der Vorfahren-Klassen im Konstruktor.

1 Wichtige Fakten

1.1 Const

In C₊₊ ist es möglich, Variablen per Deklaration unveränderlich zu machen. Das geschieht durch Voranstellen des Schlüsselwortes `const`. Man muss einer solchen Variable allerdings mit der Deklaration bereits einen Wert zuweisen.

1.2 Explicit

Mit dem C₊₊ Schlüsselwort `explicit` kann man dem Compiler implizite Typumwandlungen verbieten

1.3 Virtual

Pattern-Matching für Methoden wird erst zur Laufzeit entschieden.

1.3.1 Pure Virtual

Das sind Methoden, welche keinen Rumpf enthalten (angegeben durch `=0`). Sie müssen in einer erbenden Klasse implementiert werden, so ähnlich wie die **Interface-Klassen** in Java.