Institut für Informatik
Albert-Ludwigs-Universität Freiburg
Prof. Dr. Peter Thiemann
Dr. Philipp Klaus Krause

Sommersemester 2019

9. Programmieren in C

Abgabe bis 9. Juli, 10:15 GMT+2.

Aufgabe 0:

Erweitere das Programm aus Aufgabe 0, Blatt 8 unter Verwendung von Aufgabe 1, Blatt 7 wie folgt um Mauseingabe!

Man gehe davon aus, dass es um jeden Punkt der Ebene ein Quadrat gibt. Anfangs sind alle Quadrate schwarz. Bei einem Klick mit der linken Maustatste auf ein Quadrat soll dieses seine Farbe auf hellgrün ändern, bei einem Klick mit der rechten Maustaste auf schwarz. Bei einem Druck auf die Taste 'a' soll der Zoom so gewählt werden, dass alle Quadrate, die nicht schwarz sind, sichtbar sind.

Für diese Aufgabe ist kein Test nötig.

Aufgabe 1:

Erweitere das Programm aus Aufgabe 0, Blatt 8 unter Verwendung von Aufgabe 1, Blatt 7 wie folgt um Dateieingabe!

Es soll ein optionales Kommandozeilargument geben, das als Name einer einzulesenden Datei angesehen wird. Wenn das Argument nicht angegeben wird, sind alle Quadrate schwarz. Jede Zeile der Datei bestehe (abgesehen vom Zeilenumbruch) aus Leerzeichen und '*' (welche Quadraten entsprechen) und optionalen Kommentaren. Das erste Zeichen der Datei entspreche dem Quadrat bei $(0, 0)$, die weiteren der Zeile denen rechts davon, die weiteren Zeilen entsprechend darunter (das erste Zeichen der zweiten Zeile entspricht also dem Quadrat bei $(0, -1)$, das der dritten dem Quadrat bei $(0, -2)$, etc bis zum Ende der Datei). Leerzeichen ergeben schwarze Quadrate, '*' ergeben hellgrüne Quadrate. Falls ein Zeichen weder ein Leerzeichen, noch ein '*' ist, wird dieses, und alle folgenden Zeichen der Zeile, ignoriert.

Aufgabe 2:

Erweitere das Programm aus Aufgabe 0 oder 1 wie folgt um Dateiausgabe!

Bei Druck der Taste 's' soll eine Datei `current.life` geschreiben werden, die für jedes grüne Quadrat ein '*' enthält. Schwarze Quadrate werden als Leerzeichen dargestellt. Das erste Zeichen der Datei entspreche dem Quadrat bei (x, y) für x Rückgabewert von `intplane_get_min`, y Rückgabewert von `intplane_get_max`. Die weiteren Zeichen der Zeile entsprechen den Quadraten rechts davon, die weiteren Zeilen entsprechend den Quadraten darunter. Das erste Zeichen der letzten Zeile entspreche dem Quadrat bei (x, x) für x Rückgabewert von `intplane_get_min`.

Aufgabe 3:

Implementiere einen Stapel (ähnlich Aufgabe 2, Blatt 4), bei dem jeder Eintrag ein String oder ein `long int` ist. Das Interface besteht aus dem Typ `strs_t` und den Funktionen

```
int push_string(strs_t *strs, const char str[]);
int push_long(strs_t *strs, long int i);
int fprint_top(FILE *stream, const strs_t *strs);
int pop(strs_t *strs)
```

`push_string` legt eine Kopie eines Strings auf dem Stapel ab. `push_long` legt einen long auf dem Stapel ab. `pop` entfernt den obersten Eintrag vom Stapel (und gibt dabei soweit sinnvoll Speicher frei). `fprint_top` schreibt eine Darstellung des obersten Elements nach stream. Alle Funktionen geben bei Erfolg 0 zurück, im Fehlerfall einen Wert kleiner 0. Ein neuer Stapel soll durch Initialisierung mit `{0}` nutzbar sein. Intern soll für jeden Stapeleintrag eine `struct` genutzt werden, die eine `union` enthält, die wiederrum einen `char *` und einen `long int` enthält.

Hinweise (gelten für dieses Blatt):

- Wenn sowohl Aufgabe 1, als auch Aufgabe 2 bearbeitet wird, soll eine Datei aus Aufgabe 2, wenn sie wie in Aufgabe 1 geladen wird, (bis auf Veschiebung) wieder das gleiche Muster an schwarzen und grünen Quadraten ergeben.
- Die Aufgabenstellung bei Aufgabe 3 gibt nur vor, welche Membervariablen es in der `struct` mindestens geben muss. Es dürfen, soweit sinnvoll, auch weitere verwendet werden (z.B. Zeiger, um auf andere Einträge zu verweisen, oder eine Variable, die angibt, welcher Typ im Eintrag gespeichert ist).
- Bei Aufgabe 3 darf als Grundlage eine Lösung der Aufgabe 2, Blatt 4 verwendet und angepasst werden.
- Die Lösung von Aufgaben 0, 1 und 2 kann (abgesehen von den Tests) aus einem einzigen Programm (aber nicht einer einzigen Quelldatei) bestehen.