

Praxis der Programmierung

5. Aufgabenblatt

1. Kopieren Sie die Datei `typfehler.c` zur Behandlung von Typfehlern durch ungeeignete Benutzereingaben mittels `scanf`. Ändern Sie das Programm so, dass eine Fehlerbehandlung mit Recovering erfolgt.
2. Schreiben Sie ein C-Programm `pkub.c`, das mit einem Parameter gestartet werden muss (sonst erfolgt eine Fehlermeldung) und dann die Kubikzahl des Parameters als `double`-Zahl ermittelt und auf `stdout` ausgibt.
3. Definition einer Struktur von Punkten in der Ebene
 - a) Kopieren Sie die Datei `point_1.c` in Ihr Arbeitsverzeichnis und übersetzen Sie sie mit einem C-Compiler. Analysieren Sie den Quellcode.
 - b) Kopieren Sie `point_1.c` in `point_2.c` und editieren Sie die Kopie. Die Datei `point_2.c` soll eine Methode zum Verschieben auf neue Koordinaten enthalten. Testen Sie diese Methode. Überprüfen Sie dabei, ob beim Initialisieren einer Strukturvariable mit einer anderen (bereits existierenden) Strukturvariablen alle Member-Werte im Speicher kopiert werden oder ob nur ein Pointer übergeben wird.
 - c) Kopieren Sie die Datei `point_3.c` in Ihr Arbeitsverzeichnis und übersetzen Sie sie mit einem C-Compiler. Analysieren und erklären Sie den Quellcode.
4. Ein *Polygonenzug* ist eine zusammenhängende Folge von Strecken, der durch eine Liste von Punkten spezifiziert werden kann. Implementieren Sie den Datentyp Polygonenzug als einfach verkettete Liste von Punkten in der Ebene (aus Aufgabe 2). Dieser Datentyp soll folgende Operationen bereitstellen:
 - **append** zum Verlängern eines Polygonenzugs durch Anfügen eines neuen Punkts am Ende der Liste,
 - **shorten** zum Verkürzen eines Polygonenzugs durch Entfernen des letzten Punkts aus der Liste,
 - **pretty_print** zum Ausgeben eines Polygonenzugs durch Ausgabe seiner Punkte auf die Konsole,
 - **insert** zum Einfügen eines neuen Punkts an einer bestimmten Position (Index) in der Liste,
 - **mirror**, die einen neuen Polygonenzug erzeugt, in dem alle Punkte des übergebenen Polygonenzugs in umgekehrter Reihenfolge auftreten.

Testen Sie Ihre Implementierung, in dem Sie zunächst einen leeren Polygonenzug anlegen, dann nacheinander zwei Punkte anfügen und diese dann nacheinander wieder entfernen. Lassen Sie vor und nach jeder Operation den Polygonenzug ausgeben.