

# Arbeit mit Pointern - einfach verkettete Zyklen

## Sonderübung

Dana Liebscher

PROG Sonderübung

April 24, 2024

1. Wiederholung
2. Gefahren beim Umgang mit Pointern
3. Übung (Routenplan)

# Wiederholung

- Pointer haben das **POINTER-Attribut** (INTEGER, POINTER :: p,q)
- Pointerzuweisung mit  $=>$
- NULLIFY(p) und NULL()
- ASSOCIATED
- ALLOCATE und DEALLOCATE

# Gefahren beim Umgang mit Pointern

- Verwendung/ Dereferenzierung / Abfrage des Zustands eines Pointers in einem undefinierten Zustand
- dangling pointers: Zugriff auf bereits gelöscht Objekt (kann unter anderem passieren, wenn mehrere Pointer auf dasselbe Objekt zeigen und einer davon deallokiert wird.)
- Speicherleichen/ Memory Leaks: Wir lassen Objekte im Speicher zurück, die wir nicht mehr erreichen können

Der Compiler findet diese Fehler im Normalfall nicht!

**Ausweg:** Das Programm **valgrind**

`gfortran -g Dateiname -o Programmname`

`valgrind .\ Programmname`

# Rutenplan

Es soll ein zyklischer Rutenplan erstellt werden. Zu jedem Ort soll man eine To-DO Liste erstellen können.

Schreibe folgende ADTs:

- **todo\_task** welcher eine Aufgabe (Character, Länge 20) und einen Pointer auf eine weitere Aufgabe (*todo\_task*) enthält
- **todo\_list** welcher einen Headpointer auf eine Aufgabe (*todo\_task*) enthält
- **place** welcher den namen eines Ortes (Character, Länge 20), eine To-Do Liste (*todo\_list*), eine Zeitangabe bis zum nächsten Ort (Integer) und einen Pointer zum nächsten Ort (*place*) enthält
- **route\_cycle** welcher den Namen der Route (Character, Länge 20) und einen Pointer auf den Start der Route (*place*) enthält

Schreibe folgende Subroutinen für die To-Do Listen (Einfach verkettete Listen, siehe Aufgabe von SU02):

- **build\_todo\_list** welche eine Liste (*todo\_list*) erhält und Aufgaben (*todo\_task*) einliest und an die List hängt solange der Nutzer möchte
- **delete\_todo\_list** welche eine Liste (*todo\_list*) erhält und diese komplett löscht
- **ausgabe\_todo\_list** welche eine List (*todo\_list*) vollständig kommentiert ausgibt

Schreibe folgende Subroutinen für den Routenplan unter Nutzung der obigen Subroutinen (Einfach verketteter Zyklus):

- **build\_route** welche eine Route (*route\_cycle*) erhält, und diese mit Orten (*place*) füllt solange der Nutzer möchte. Außerdem sollen immer die Zeitangabe bis zum nächsten Ort und eine evtl. To-Do Liste abgefragt werden
- **delete\_route** welche eine Route (*route\_cycle*) inklusive aller To-Do Listen komplett löscht
- **ausgabe\_route** welche eine Route (*route\_cycle*) inklusive aller To-Do Listen komplett kommentiert ausgibt

Außerdem soll eine Funktion **laufzeit** geschrieben werden, welche die komplette Laufzeit (alle Zeitangaben bis zu den nächsten Orten) berechnet. Zusätzlich kann man Zeitangaben zu den To-Do Listen hinzufügen und diese hier mit einrechnen.