

2. Übung zu TEI2_CPR

1. Zeiger

- Definieren Sie: `float f=17;`
- Lassen Sie einen Zeiger `fp` auf `f` zeigen.
- Benutzen Sie `fp`, um die 17 auszugeben.
- Bei `printf` steht `%p` für die Ausgabe einer Adresse. Lassen Sie sich die Adresse von `f`, den Zeiger `fp` und die Adresse von `fp` ausgeben.
- Reservieren Sie Speicher für einen `float`-Wert *auf dem Heap*. Die hierzu notwendige Variable soll `fheap` heißen. Setzen Sie den `float`-Wert auf 5.0.

2. Funktion „quadriere“

- Schreiben Sie eine Funktion `void quadriere(„Datentyp“ x)`, die den über `x` erreichbaren Wert durch sein Quadrat *ersetzt* (für „Datentyp“ etwas Sinnvolles¹ einsetzen!).
- Wenden Sie die Funktion auf `f` und `fheap` an, lassen Sie die Ergebnisse ausgeben und kontrollieren Sie sie.
- Geben Sie, wo nötig, den Speicher wieder frei.

3. Zeiger und Strukturen

- Definieren Sie eine Struktur `Vektor2D` mit den `float`-Member-Variablen `x` und `y`.
- Legen Sie *je eine* solche Struktur auf dem Stack und auf dem Heap an.
- Setzen Sie bei beiden Strukturen die jeweilige Member-Variable `x` auf 4.
- Benutzen Sie die Ihre oben definierte Funktion `quadriere` um die jeweilige Member-Variable `x` zu quadrieren. Lassen Sie die Ergebnisse ausgeben und kontrollieren Sie sie.
- Geben Sie, wo nötig, den Speicher wieder frei.

¹ Typ von `x` soll aber *keine* sog. Referenz sein. Referenzen gibt es nur in C++ und hatten wir noch nicht in der Vorlesung. (Referenzen werden mit `&` angelegt – wenn Sie davon noch nichts gehört haben, umso besser: dann diese Fußnote komplett ignorieren!). Insbesondere soll die Signatur **nicht** `void quadriere(float&)` sein.