Praktikum II
Embedded Software

Prof. Dr. Nikolaus Wulff

02. November 2020

1 Generierung objektbasierter „Klassen“

In letztem Prakikum wurden die ersten Schritte mit einer objektbasierten Entwicklung in C gemacht. Vieles wurde noch händisch durchgeführt – teilweise mit Copy & Paste und Codeduplizierung, sowie nachfolgendem Suchen & Ersetzen, um die Typnamen der Strukturen zu ändern.

Ein auf Dauer unbefriedigendes und fehlerträchtiges Vorgehen. In diesem Praktikum geht es darum, die in der Vorlesung besprochenen C Makros zu implementieren, auszuprobieren und zu verfeinern.

Aufgabe

- Erstellen Sie in einer Headerdatei, z.B. `CObjects.h` geeignete Makros, die basierend auf den C Präprozessoranweisungen „##“ und „#“ textuelle Ersetzungen zur Codegenerierung vornehmen. Hierbei soll das in der Vorlesung behandeltete architektonische Muster der *kanonischen Zerlegung* in zwei Strukturen für Instanzen und Klassen erzeugt werden. Die Makros sollen ähnlich zu C++ Templates oder Java Generics für beliebige Typen <T> passende Strukturen und `typedef`s etc. generieren.

- In der finalen Endausbaustufe sind die Makros so mächtig, dass die Headerdateien in Listing (1) und (2) ausreichen, um den notwendigen Code für Klassen- und Instanzenstrukturen des Shape Beispiels unter Einbeziehung der Vererbungsbeziehung generieren zu lassen. Die C Implementierungen müssen allerdings leider – wie gehabt – noch händisch selbst ausprogrammiert werden.

Hinweis

Es mag hilfreich sein sich den vom C Präprozessor generierten Code anzuschauen. Der `gcc` Compiler lässt sich mit dem Flag „-save-temps“ dazu

bewegen, die Zwischenergebnisse auf Platte zu speichern, so dass diese lesbar werden, was die Fehlersuche in den Makros und dem generierten Code deutlich erleichtert.

```
#ifndef CIRCLE_H_
#define CIRCLE_H_
#include "CObjects.h"
#include "Shape.h"

#define Circle_METHODS          \
  void (*resize) (Circle obj, int r);

#define Circle_ATTRIBUTES       \
  int r;

BEG_DEFINE_CLASS(Circle) EXTENDS(Shape)
  METHODS(Circle)
END_DEFINE_CLASS(Circle)


BEG_DEFINE_INSTANCE(Circle) INSTANCE_OF(Shape)
  ATTRIBUTES(Circle)
END_DEFINE_INSTANCE(Circle)

#endif /* CIRCLE_H_ */
```

Listing 1: Headerdatei des Shape Beispiels für einen Kreis.

Auch `DEFINE_CLASS` und `DEFINE_INSTANCE` lassen sich in einem Makro zusammenfassen:

```
#ifndef RECT_H_
#define RECT_H_
#include "CObjects.h"
#include "Shape.h"

#define Rect_METHODS      \
  void (*resize) (Rect r, int w, int h);

#define Rect_ATTRIBUTES   \
  int w;                  \
  int h;

DEFINE_EXTENDED_TYPE(Rect,Shape)

#endif /* RECT_H_ */
```

Listing 2: Kürzer lässt sich eine Vererbungsbeziehung nicht codieren.