

Objektorientierte Modellierung, Spezifikation und Implementierung (OMSI) I

Praktikum

Prof. Joachim Fischer / Dr Klaus Ahrens

Dr. Markus Scheidgen

{fischer,ahrens,scheidge}@informatik.hu-berlin.de

LFE Systemanalyse, III.310

Aufgaben zur Veranstaltung

- ▶ <http://www.informatik.hu-berlin.de/forschung/gebiete/sam/Lehre/omsi>
- ▶ Fragen: per Mail über Goya

Aufgabe 2 – Eine Uhr mit zeigerbasierter Alarmverwaltung in C++

- ▶ Üben der folgenden C++ Konzepte
 - Headerdateien
 - (einfache) Klassen und Instantiierung
 - Zeiger und Felder
- ▶ In einer weiteren Aufgabe als Event-basierte Simulation
- ▶ In einer noch weiteren Aufgabe als ODEM-x Simulation

Aufgabe 2 – Eine Uhr mit zeigerbasierter Alarmverwaltung in C++

- ▶ Schreiben Sie Headerdateien und Implementierungen für die Klassen *Clock* und *Alarm*.
- ▶ Die Implementierung soll die gewünschte Ausgabe erzeugen.

```
#include <stdlib.h>
#include "clock.h"
#include "alarm.h"

using namespace aufgabe2;

int main(void) {
    Clock* clock = new aufgabe2::Clock();

    clock->setAlarm(new Alarm(12,43,0));
    clock->setAlarm(new Alarm(12,11,0));
    clock->setAlarm(new Alarm(12,17,0));

    clock->run();

    return EXIT_SUCCESS;
}
```

```
ALARM: 12:11:0
ALARM: 12:17:0
ALARM: 12:43:0
```

Aufgabe 2 – Eine Uhr mit zeigerbasierter Alarmverwaltung in

C++

► Anforderungen:

- Alarme werden nie kopiert!
- Automatische Felderweiterung für beliebig viele Alarme
- Bubblesort (oder schneller) zur Sortierung der Alarme
- Baldige Speicherfreigabe
- keine Bibliotheken verwenden
- Makefile!

```
#include <stdlib.h>
#include "clock.h"
#include "alarm.h"

using namespace aufgabe2;

int main(void) {
    Clock* clock = new aufgabe2::Clock();

    clock->setAlarm(new Alarm(12,43,0));
    clock->setAlarm(new Alarm(12,11,0));
    clock->setAlarm(new Alarm(12,17,0));

    clock->run();

    return EXIT_SUCCESS;
}
```

```
ALARM: 12:11:0
ALARM: 12:17:0
ALARM: 12:43:0
```

Punkte

- ▶ Aufgaben werden, wenn nicht anders angegeben bewertet.
- ▶ Für jede Abgabe werden in Goya Punkte vergeben.
- ▶ 50% der maximal zu erzielenden Punkte müssen erreicht werden, um zur Prüfung zugelassen zu werden.
- ▶ Für die erste Aufgabe werden maximal 3 Punkte vergeben.
 - kompiliert (2), Testlauf erzeugt keine Fehler (1)
- ▶ Für die zweite Aufgabe werden maximal 8 Punkte vergeben.
 - kompiliert und richtige Ausgabe (3), Schnittstellen (1), Headernutzung (1), Bubblesort (1), Felderweiterung (1), Speicherfreigaben (1)

▶ nächstes Praktikum in 2 Wochen

Fragen?