

Objektorientierte Programmierung mit C++ (WS 2010)

Dr. Andreas F. Borchert, Tobias Brosch
Institut für Angewandte Informationsverarbeitung
Universität Ulm
Blatt 7: Abgabetermin 08. Dezember 2010

Dynamic Function Registry

Aufgabe: Implementiert die `DynFunctionRegistry` analog zu den Folien aus der Vorlesung.

Die Datei `main.C`, soll dabei die Eingabeschleife enthalten, die automatisch versucht eine Funktion zu laden, wenn diese noch nicht in der Registry enthalten ist. Die abstrakte Klasse `Function`, soll hierbei in `Function.h` definiert werden, die Klasse `DynFunctionRegistry`, in den Dateien `DynFunctionRegistry.hC`.

Implementiert drei Funktionen in jeweils separaten Klassen: `Sinus`, `Cosinus` und `Exp` (jeweils Header und Source-File). Diese sollen im Ordner `pluginFuncs` abgelegt werden, und in den shared-libraries `libsins.so`, `libcos.so` und `libexp.so` zur Verfügung gestellt werden (ebenfalls im Ordner `pluginFuncs`).

Einreichen der Lösung mit:

```
submit cpp 7 myLsgB07.tgz
```

Dabei soll das Archiv `myLsgB07.tgz` die erwähnten Dateien, sowie den Ordner `pluginFuncs` mit seinen Dateien enthalten. Um das Beispiel zu kompilieren, soll *ein* Aufruf von `make` ausreichend sein (Ihr könnt gerne auch ein eigenes Makefile für den Ordner `pluginFuncs` schreiben, das in dem eigentlichen Makefile mitverwendet wird). Bitte *keine* weiteren Dateien wie z.B. Object-Files!

Hinweise

- Eine shared-Library kann mit

```
g++ -shared -o libLibraryName objectFiles
```

gebaut werden.
- Die Object-Files sollten mit der Option `-fPIC` kompiliert werden. Dies sorgt dafür, dass Position-Independent Code erzeugt wird.

Viel Spaß!