

Esercitazione 10

18/02/2013

Esercizio

Si vuole integrare nella programma `AllDynamic` la possibilità di gestire anche le integrande attraverso le factory. In particolare si vuole collezionare una serie di librerie dinamiche contenenti le integrande, anche definite tramite funtori. Il caricamento delle librerie, il cui numero è a priori incognito, deve rimanere dinamico e la scelta dell'integranda deve poter essere fatta ancora tramite il file di dati gestito da `GetPot`. Ogni libreria che contiene le funzioni deve essere indipendente dalle altre e deve poter autonomamente registrare nella factory i propri dati. In questo modo all'aggiunta di una nuova libreria non bisognerà fare altro che inserire il nome della nuovo libreria nel file di dati, i suoi oggetti verranno ad essere automaticamente gestiti della factory.

Non si vuole modificare la classe `Factory` in alcun modo. I punti per ottenere quanto richiesto sono i seguenti, facendo attenzione di modificare opportunamente in `Makefile` per poter gestire le nuove funzionalità.

1. Non utilizzare più la classe `udfHandler`, al suo posto verrà utilizzata la factory.
2. Utilizzare il function wrapper della standard library per uniformare il tipo che dovrà essere contenuto nelle factory.
3. Creare un nuova nuova classe proxy che servirà per la registrazione delle funzioni. Capire perché è utile incorporare il metodo builder dal proxy vero e proprio.
4. Non è quindi più utile inserire le funzioni all'interno degli `extern C`. Creare quindi un file per la registrazione delle funzioni utilizzando in proxy introdotto.
5. Creare un funtore e associargli la registrazione.

Un'aggiunta per rendere più robusta la gestione delle regole di quadratura è determinare staticamente se il parametro `AbstractProduct_type`, definito nella factory è genitore di `ConcreteProduct`. Inoltre il metodo statico `Build` deve essere poter convertito nel tipo corrispettivo definito nella factory. Utilizzare i `traits` definiti nella standard libray e lo `static_assert`.