# Common Patterns

## Singleton (Unique)

Sometimes you need to get an instance of a certain Subject that has a globally unique combination of Traits. This can be thought of something like a *singleton* pattern within a classical OOP approach.

You can totally achieve that with the help of a simple method like this:<code cpp> template < typename T, typename ...Ts > T GetSingleton(AMechanism* const Mechanism) {

```
const auto Filter = FFilter::Make<T, Ts...>();
const auto Chain = Mechanism->Enchain(Filter);
```

```
auto Cursor = Chain->Iterate(0, 1);
if (!Cursor.Provide())
{
    checkNoEntry()
    return T();
}
const auto Trait = Cursor.GetTrait<T>();
verifyf(!Cursor.Provide(), TEXT("Two singleton objects detected!"));
return Trait;
```

} </cpp>

From:
http://turbanov.ru/wiki/ - **Turbopedia**

Permanent link:
**http://turbanov.ru/wiki/en/toolworks/docs/apparatus/patterns?rev=1651658331**

Last update: **2022/05/04 09:58**