



Branch: master ▾

Find file

Copy path

[cpp_kursu_odevleri](#) / [templates](#) / templates_10.md

Fetching contributors...



32 lines (24 sloc) | 1.23 KB

Raw

Blame

History



Aşağıda yer alan kodda, bir sınıfın *begin* isimli bir üye işleve sahip olup olmadığını "*expression sfinae*" tekniği ile sınavan *has_member_begin* isimli bir sınıf şablonu yer alıyor:

```
#include <type_traits>

struct has_member_begin_test
{
    template<class U>
    static auto test(U* p) -> decltype(p->begin(), std::true_type());

    template<class>
    static auto test(...) -> std::false_type;
};

template<class T>
struct has_member_begin
: decltype(has_member_begin_test::test<T>(0)) {};
```

- Öncelikle yukarıdaki kodu inceleyerek anlamaya çalışın.
- Aşağıdaki parametrik yapıda bir işlev şablonu yazacaksınız:

```
template<typename T>
void func(const T &r)
{
    //code
}
```

has_member_begin isimli *trait* sınıfından faydalanarak *func* işlev şablonundan yalnızca *begin* isimli bir üye işleve sahip *T* türleri için işlevler üretilmesini sağlamanız gerekiyor. Eğer *T* türü *begin* isimli bir işleve sahip değil ise yazacağınız *func* işlev şablonunun işlev yükleme çözümlemesine (function overload resolution) katılmaması gerekiyor. *SFINAE* tekniklerinden birini kullanarak *func* işlev şablonunu gerçekleştirin. Yazdığınız şablonu farklı türler için test edin.

© 2020 GitHub, Inc.

- Terms
- Privacy
- Security
- Status
- Help

- Contact GitHub
- Pricing
- API
- Training
- Blog
- About