

extra11. Реализуйте метафункцию `is_class<T>`, позволяющую проверить, является ли тип `T` классом или структурой. Что можно сказать о реализации `is_union<T>`?

Реализация `is_class<T>`:

```
1
2 namespace detail {
3     template <class T> char test(int T::*);
4     struct two { char c[2]; };
5     template <class T> two test(...);
6 }
7
8 template <class T>
9 struct is_class : std::integral_constant<bool, sizeof(detail::test<T>(0))==1
    && !std::is_union<T>::value> {};
```

Пример использования:

```
1
2 struct A {};
```

```
3
4 class B {};
```

```
5
6 enum class C {};
```

```
7
8 int main()
9 {
10     std::cout << std::boolalpha;
11     std::cout << std::is_class<A>::value << '\n';
12     std::cout << std::is_class<B>::value << '\n';
13     std::cout << std::is_class<C>::value << '\n';
14     std::cout << std::is_class<int>::value << '\n';
15 }
```

Основные мысли:

1. Используем SFINAE и радуемся жизни. [тык почитать поподробнее](#)
2. Class и struct вообще в c++ неразличимы.