Вопрос 25

Реализация с cppreference (ну почти)

```
#include <type_traits>

namespace detail {

template <typename T>
auto detect_is_polymorphic(int) ->
     decltype(dynamic_cast<const void*>(std::declval<T*>()), std::true_type());

template <typename...>
std::false_type detect_is_polymorphic(...);

}

template <class T>
struct is_polymorphic :
    std::bool_constant<std::is_class_v<T> &&
        decltype(detail::detect_is_polymorphic<T>(0))::value> {};
```

Реализация работает так: класс T является полиморфным, если T* можно динамически привести к const void*. Эта идея и используется в реализации. Если dynamic_cast возможен, то будет выбрана первая версия (и возвращаемый тип - std::true_type). Если нет, то будет Substitution Failure, и благодаря SFINAE будет выбрана вторая версия с возвращаемым типом std::false_type.