

УПРАЖНЕНИЯ

Решение задач можно присылать (предпочтительнее ссылка на github, и т.п.) на vkonoovodov@gmail.com. Бонусные баллы за решения задач могут быть поставлены нескольким первым приславшим правильное решение. Решение должно быть оформлено в виде компилирующегося кода.

- (1) Используя идиому Type Erasure, напишите класс `Any`, сохраняющий объекты любого типа, и обеспечивающий к ним доступ:

```
Any a(5);  
a.get<int>(); // 5  
a.get<std::string>(); // error
```

- (2) Определите структуру

```
template<class...>  
struct are_same;
```

которая содержит поле `value` со значением `true` в случае, когда все параметры шаблона имеют один и тот же тип без учета квалификаторов и ссылочности, и `false` иначе.

Воспользуйтесь `std::true_type`, `std::false_type`, `std::is_same` и механизмом наследования. Определите также `constexpr bool are_same_v` для быстрого доступа к этому значению:

```
static_assert(are_same_v<int, int32_t, signed int>, "compile assert");
```

- (3) Определите класс `TMyException`, унаследовав его от одного из стандартных исключений, определите в нем оператор `<<` для записи в объект исключения сопровождающего текста при его генерации. Породите из него несколько классов-наследников. Продемонстрируйте перехват исключений этого типа с помощью теста `EXPECT_THROW`.