

## ДОМАШНЯЯ РАБОТА ПО ПРОГРАММИРОВАНИЮ НА C++ №1

Решения принимаются до 8 марта включительно.

**Задача 1** (6 баллов). Реализуйте шаблонный класс `SmartPointer`, представляющий собой «умный» указатель. Класс должен иметь три шаблонных параметра:

- тип `T`, на который ссылается указатель;
- тип `StorageStrategy`, определяющий способ хранения и удаления ресурса;
- тип `OwnershipStrategy`, определяющий политику владения ресурсом (копирования и присваивания «умных» указателей).

Предоставьте две стратегии хранения (удаления): указатель на отдельный элемент (по умолчанию) и массив. Предоставьте также как минимум четыре стратегии владения:

- копирование запрещено (по умолчанию);
- подсчет ссылок с помощью отдельного счетчика;
- подсчет ссылок с помощью двусвязного списка;
- передача владения при копировании (как в `std::auto_ptr`).

Классы стратегий должны предоставлять соответствующие псевдонимы, статические функции и т.д., которыми может пользоваться класс `SmartPointer`.

Продемонстрируйте работу созданного шаблонного класса. В каких случаях его можно использовать со стандартными контейнерами?

**Задача 2** (4 балла). Определите собственный класс исключения `CastException`, унаследовав его от одного из стандартных исключений. Определите в нем оператор `<<` для записи в объект исключения сопровождающего текста при его генерации. Выведите из него несколько классов-наследников. Создайте шаблонную функцию

```
template <typename T, typename CharType>
inline T FromString(const CharType* data, size_t len);
```

которая преобразовывала бы строку к типам `char`, `int`, `long`, `bool` и генерировала бы определенные выше исключения разных типов в случае, когда это невозможно (неверный символ, арифметическое переполнение и т.д.). Продемонстрируйте работу этой шаблонной функции и перехват исключений.