

## Задачи, ЗАНЯТИЕ 6

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

- (1) Напишите аллокатор, который выполняет стандартное распределение памяти с помощью `operator new` и логирует сообщения о работе каждой функции. Проверьте его работу с несколькими стандартными контейнерами. Модифицируйте аллокатор так, чтобы он по возможности выделял блоки памяти из внутреннего статического буфера.
- (2) Реализуйте указатель `TIntrusivePtr`, создав базовый класс `TBasePtr`, в котором определите операторы `->`, `*`, `==`, `!=`, `bool` (предполагая, что отнаследовав от этого класса, можно реализовать и другие умные указатели). В основном классе должны быть определены основные копирующие и перемещающие операции, методы `UseCount`, `Get`, `Reset`, `Release`, а также конструктор от обычного указателя. Указатель `TIntrusivePtr` использует механизм встроенного подсчета ссылок (счетчик ссылок находится не в самом указателе, как в `std::shared_ptr`, а в объекте).

Продемонстрируйте работу умного указателя с каким-нибудь счетчиком ссылок.

Использование может быть, например, таким:

```
class TDoc: public TRefCounter<TDoc> {
    using TPtr = TIntrusivePtr<TDoc>;
    // ...
}
```

Обратите внимание, что в случае таких указателей конструкция

```
TDoc* p = new TDoc;
TIntrusivePtr<TDoc> ip1(p);
TIntrusivePtr<TDoc> ip2(p);
```

вполне безопасная.