

Результат выполнения задания необходимо отправить по электронной почте [job@polator.ru](mailto:job@polator.ru), в одном ZIP архиве, без полных и промежуточных файлов. Не забудьте указать ваше ФИО или иные контакты, чтобы мы вас смогли найти.

Задание состоит из единого текста задачи и трех возможных уровней его исполнения (всего 2 страницы). Уровни отличаются между собой глубиной проработки, по нарастающей. Для выполнения старших по номеру Уровней, требуется выполнение всех предыдущих. Для минимального выполнения задания, достаточно выполнить только первый Уровень. Последующие Уровни (второй и третий) являются дополняющими, но первый – обязательный. Таким образом, первый уровень исполнения задания является проходным на конкурсе, а последующие уровни будут демонстрировать ваше профессиональное конкурентное преимущество - дополнительно.

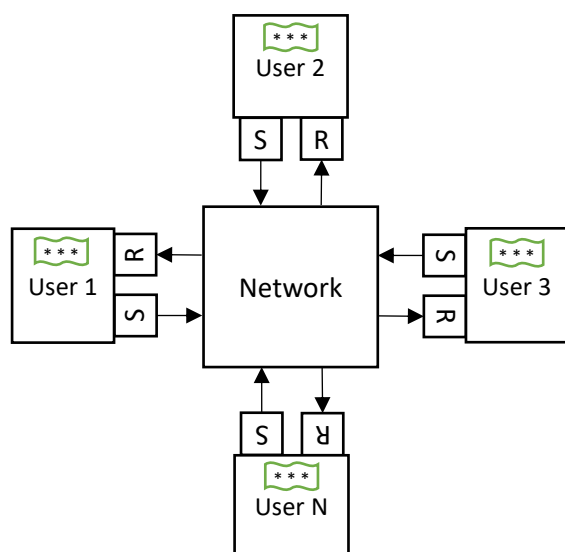
В задании не требуется реализовывать сетевое LPC/RPC взаимодействие (например HTTP, Socket, Mailslot). Так же не требуется разработка сетевой версии протокола обмена информацией (например XML или JSON). В спорных случаях просим использовать MOCK объекты и структуры для DirectCall обмена данными. И, так как целью задания не является создание решения на какую-либо конкретную программную платформу, то для выполнения задания можно использовать любую доступную среду программирования и отладки (Windows/Linux), любых доступных вам версий. Например, нам доступны MS Visual Studio 2019 и GCC 9, включая x64 тулчейны. Вы можете использовать иные.

### Задание

Создать прототип социальной сети обмена текстовыми SMS-сообщениями между ее пользователями (под текстовым сообщением понимается текст в формате ASCII).

Типовой унифицированный пользователь **User** может зарегистрироваться в сети **Network**, подписаться на других пользователей, получать от них текстовые SMS сообщения, а также отправлять текстовые SMS сообщения своим подписчикам. При исполнении задания любого из уровней, эти два объекта (User и Network) должны присутствовать обязательно. Остальные введенные объекты являются предметом исполнения задания конкретным исполнителем, т.е. демонстрируют его взгляд на ООД.

У пользователя должна присутствовать лента, содержащая все его сообщения, а также сообщения всех пользователей, на которых он подписан.



Обязательный набор функций, которые требуется спроектировать, должен содержать:

- Зарегистрироваться в сети
- Подключиться к сети
- Получить список зарегистрированных пользователей
- Получить список подключенных пользователей
- Подписаться на пользователя
- Послать пользователю сообщение

Дополнительные функции могут быть реализованы по усмотрению исполнителя.

## Уровень 1

(временная сложность 2-6 часов)

Уровень тестирует навыки объектного моделирования и дизайна (т.е. декомпозиция объектов, раздача обязанностей включая методы и данные, архитектура используемого паттерна взаимодействия/коммуникации и тд), включая технику исполнения на языке C++, т.е. требуется продемонстрировать ООД самого решения. Обратите внимание – требуется объектно-ориентированный дизайн!

Результатом на проверку является набор, т.е. один или более заголовочных файлов на языке C++ (которые могут быть использованы директивой `include` в стороннем проекте), не использующий сторонние библиотеки. Стандартные, а также дефакто, например `boost`, разрешается использовать.

Весь ваш код со всеми объектами и их методами должен быть в заголовочных файлах, т.е. не разделен на файлы `.cpp` и файлы `.h`, `.hpp`.

## Уровень 2

(временная сложность 2-3 дня, включая Уровень 1)

Если рассматривать полученный результат Уровня 1, как результат, при котором создается SDK, то Уровень тестирует умение писать Unit-тесты на все возможности, заложенные в этом SDK, т.е. `Compliance Test`, включая технику исполнения на языке C++.

Допускается использование Unit-тест фреймворков, а также платформ зависимых библиотек или функций (например, `threading` и тд), но таким образом, чтобы не происходило модификации исходных текстов Уровня 1, т.е. SDK должен подключаться директивой `include` “как есть”.

Результатом на проверку является компилируемый исходный текст на языке C++, т.е. исполнимый файл, в котором проверяется работоспособность SDK.

## Уровень 3

(временная сложность 1 неделя, включая Уровень 1 и 2)

Задание тестирует навыки создания работоспособного прототипа будущего решения, где необходимо создать пользовательский интерфейс к ранее спроектированному (Уровень 1) и оттестированному (Уровень 2) SDK, демонстрирующий этот прототип обычному пользователю на одном персональном компьютере, в одном процессе.

Сам пользовательский интерфейс может быть как текстовый, так и графический, и может использовать любые сторонние библиотеки по усмотрению разработчика, но таким образом, чтобы не происходило модификации исходных текстов Уровня 1, т.е. SDK должен подключаться директивой `include` “как есть”.

Результатом на проверку является компилируемый исходный текст на языке C++, т.е. исполнимый файл, который выводит на экран 4 клиентских окна для демонстрации возможностей прототипа обычным пользователем.

Если потребуются разъяснения или консультации, то просим внимательно изучить текст задания выше. Если ответа не удалось найти, то можно задать вопрос по адресу [job@polator.ru](mailto:job@polator.ru)

После выполнения уровня задания (первого, первого + второго, или первого + второго + третьего), не забудьте указать ваше ФИО, а так же приложить к письму все ваши материалы, в одном ZIP архиве, без исполнимых и промежуточных файлов.