

Messaging. Complex Task

Примечание. Это задание является дополнением к основному практическому заданию. Без выполненного общего задания, *Complex Task* не засчитывается.

Задание

Ниже описан механизм централизованного контроля и управления, расширяющий реализованную ранее систему.

Что нужно сделать

1. Создайте UML схемы для реализованного вами ранее решения (дорабатывать саму программу не обязательно):
 - Component Diagram
 - Sequence Diagram.
2. Отобразите на схеме недостающие элементы так, чтобы она также отражала принцип действия описанного ниже механизма.
3. Обговорите с ментором получившуюся схему, обсудите детали:
 - a. какие механизмы очередей нужно задействовать, чтобы реализовать недостающую часть системы?
 - b. как поведет себя система в случае временного выхода из строя определенных сервисов (например - перезагрузка брокера сообщений/центрального сервера/клиентской службы, во время отправки/получения сообщений)?

Механизм централизованного контроля и управления

Механизм позволяет серверу получать от служб информацию об их текущем статусе и при необходимости - рассылает службам новые настройки.

После того, как служба получает новые настройки, она применяет их ко всем новым сообщениям.

В рамках этого механизма:

- От служб ввода на центральный сервер с некоторой периодичностью приходит текущий статус службы:
 - Чем занята служба (ждет новых файлов/обрабатывает последовательность/...)
 - Текущие настройки:
 - Максимальный размер пересылаемого сообщения
 - Какие-либо дополнительные настройки (если есть необходимость)
- От сервера ко всем службам уходят команды:

- Обновить статус (т.е. не дожидаясь времени обновления статуса)
- Поменять настройки:
 - Максимальный размер пересылаемого сообщения
 - Какие-либо дополнительные настройки (если есть необходимость)

Links:

1. [Component diagrams.](#)
2. [Message queue example in component diagrams.](#)
3. [Sequence diagrams.](#)

<http://draw.io/> - один из инструментов для создания UML, approved freeware.