

## Задание 4. Чат

Необходимо написать клиент-серверное приложение – чат с ui-интерфейсом клиента.

### Общие требования:

1. Реализовать нужно и клиент и сервер.
2. Классы клиента и сервера должны находиться в одном проекте, но в разных пакетах.
3. Клиент и сервер не должны зависеть друг от друга, но могут зависеть от одних и тех же классов (например, от пакета common).
4. Клиент и сервер должны запускаться независимо (каждый имеет свой собственный метод main).
5. Клиент и сервер должны собираться в разные jar-архивы. Архивы клиента и сервера не должны содержать классов, не относящихся к клиенту и серверу соответственно.

### Требования к клиенту:

1. Клиент устанавливает соединение с сервером по указанному адресу.
2. Перед подключением к серверу, пользователю предлагается ввести имя (ник). С этим именем клиент подключается к серверу.
3. Клиент помечает все свои сообщения датой отправки.
4. Клиент имеет графический интерфейс, состоящий из панели вывода сообщений, панели ввода сообщения, панели со списком участников чата, а также кнопки отправить.
5. Интерфейс должен предоставлять возможность ввода адреса сервера и имени в чате, например, при старте.
6. При закрытии окна интерфейса клиента, соединение с сервером должно корректно закрываться.

### Требования к серверу:

1. Сервер ждет подключения клиентов на заданном порту (задается в конфигурационном файле).
2. При подключении клиента, сервер должен отправлять в чат сообщение, что клиент <имя> присоединился к чату.
3. При получении сообщения от клиента, сервер рассыпает это сообщение всем подключенными клиентам.
4. Каждое сообщение должно помечаться именем клиента, от которого оно пришло (это может делать как сервер при получении, так и клиент при отправке).
5. Сервер не дает подключиться клиенту с именем, которое уже занял другой клиент в чате.
6. При отключении клиента, сервер должен отправлять в чат сообщение, что клиент <имя> отключился.

## Задание 4.1. UI-интерфейс клиента

Необходимо реализовать UI-интерфейс клиента:

1. UI-интерфейс сразу должен содержать все элементы, указанные в требованиях:
  - a. должны запрашиваться адрес сервера и имя пользователя;
  - b. при отправке сообщения, оно должно отображаться в панели вывода сообщений;
  - c. в панели со списком участников должно отображаться введенное имя пользователя.
2. Необходимо сразу предусмотреть отдельный пакет для «модели» в терминах MVC. В модели должна быть сразу предусмотрена вся функциональность, которая будет реализована в будущем:
  - a. подключение к серверу (предусмотреть, что сервер может быть недоступен или возвращать ошибки, например, что ник уже занят);
  - b. отправка сообщения на сервер;
  - c. получение сообщений от сервера.

## Задание 4.2. Протокол взаимодействия сервера и клиента

Необходимо реализовать протокол взаимодействия между клиентом и сервером:

1. протокол должен содержать все типы сообщений, которые предусмотрены заданием (подключение к серверу, подключение/отключение других пользователей, обмен сообщениями пользователей);
2. протокол должен содержать все необходимые данные, предусмотренные заданием (например, дата отправка сообщения).

## Задание 4.3. Реализация сервера и взаимодействия по сети

Необходимо реализовать сервер и реальное взаимодействие клиента с сервером.