

## Задание «Пропускная способность компьютерной сети»

**Загружать решения в виде Zip архива в проект Smart LMS под названием HW1.  
В архиве должна быть папка с проектом Java и файл XLSX с графиками (см. ниже).  
Название Zip архива должно совпадать с Вашей фамилией.**

**Мотивация.** Если Ваше программное обеспечение попадает в новую компьютерную сеть, в первую очередь следует выполнить измерения характеристик новой коммуникационной среды. В первую очередь, важно знать скорость обмена данными по сети именно для Вашего конкретного приложения и бизнес-протокола. Дополнительно мотивация иллюстрировалась на семинаре 26 Jan 2024 (см. видео и слайды).

**Плагиа́т будет оценен в 0 баллов без возможности пересдачи.**

### Содержание задания.

Реализовать клиент-серверное приложение на Java (и сервер и клиент должны быть реализованы на Java). Использовать класс Socket на стороне клиента и класс ServerSocket на стороне сервера.

Клиент имеет следующие параметры командной строки: `<IP> <port> <N> <M> <Q>`. Клиент должен подключаться к серверу по IP адресу `<IP>` и порту `<port>`. Клиент должен выполнить `M` итераций и `Q` под-итераций отсылки/приема данных. На итерации номер `K=0,...,M-1` клиент отправляет серверу массив байт длиной `N*K+8` несколько раз, а именно `Q` раз. Например, `N = 32, M = 145, Q = 10`. Клиент каждый раз при отсылке формирует массив из случайного набора байт (чтобы предотвратить его кэширование на каких-либо этапах). Также следует установить флаг `TCP_NO_DELAY`, чтобы каждый массив отправлялся сразу. Клиент перед очередной отсылкой массива должен дожидаться ответа от сервера. Клиент замеряет время, которое требуется для того, чтобы отправить массив байт и дождаться ответа от сервера. Для итерации `K` требуемое время рассчитывается усреднением времени за `Q` под-итераций.

Таким образом, на итерации номер 0 клиент отправляет серверу 8 байт, на итерации 1 отправляет 32+8 байт, ..., на итерации `M-1` отправляет `32*(M-1)+8` байт. Внимание: массив должен отправляться одним вызовом метода `write` у сокета клиента.

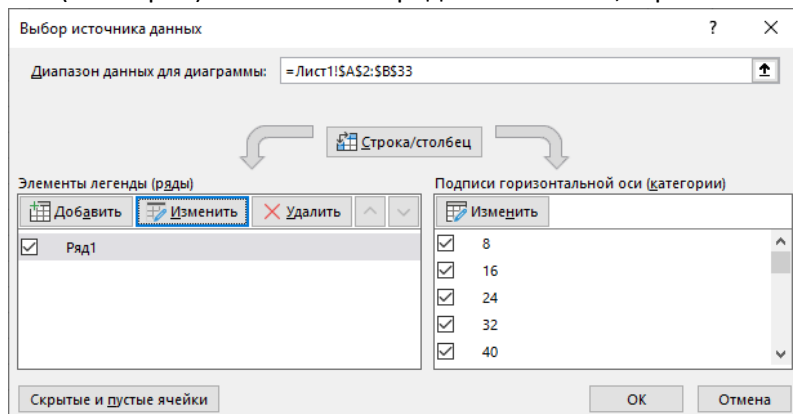
Сервер в ответ должен отправлять строку `YYYY.MM.DD HH:MM:SS` (текущая дата и время; можно добавить временную зону; можно использовать класс `Date` и формировать ответ вручную либо `DateTimeFormatter`).

Клиент не выводит в консоль ответы от сервера. Клиент замеряет для каждого запроса время, которое прошло от вызова метода `write` до получения ответа от сервера (`System.currentTimeMillis/System.nanoTime`). Для каждого значения `K` клиент рассчитывает среднее время. В конце работы клиент выводит в файл либо на консоль таблицу (число байт итерации `K`, среднее время). По этим данным необходимо построить график в Excel (горизонтальная ось – число байт итерации `K`, вертикальная – среднее время). Пожалуйста, интерпретируйте полученный график.

Количество отправляемых байт от клиента к серверу может расти до 8KB. При этом `N = 32, M = 145, Q = 10` – не единственно возможная конфигурация, при которой можно проследить интересующие нас закономерности, проведите эксперименты с разными входными параметрами, напр., `N = 1` либо 8 либо 64, `M = 200` либо 100, `Q = 1` либо 25 и так далее и постройте минимум 3

графика для разных комбинаций параметров (один из графиков должен быть для  $N = 32$ ,  $M = 145$ ,  $Q = 10$ ).

При построении графика, обратите внимание – «Элементы легенды (ряды)» - это время, вертикальная ось: нужно выбрать столбец с измеренным временем, а «Подписи горизонтальной оси (категории)» - это число передаваемых байт, горизонтальная ось.



### Указания.

Обязательно выполнить отслеживание пакетов сервера и клиента в WireShark (достаточно найти несколько пакетов и понять, что они сгенерированы Вашими клиентом и сервером).

Необходимо использовать на стороне клиента

```
Socket s = new Socket("127.0.0.1", server_port); -- IP надо будет заменить
socket.setTcpNoDelay(true); -- попробовать и с true и с false
s.getOutputStream();
```

на стороне сервера:

```
ServerSocket ss = new ServerSocket(server_port);
Socket s = ss.accept();
s.getInputStream();
```

### Развертывание клиента и сервера.

1. Во время разработки и основного тестирования, сервер и клиент должны располагаться на Вашей рабочей машине. В этом случае время лучше измерять в наносекундах; в последующих случаях, возможно, придется измерять время в миллисекундах.
2. Во время проведения полномасштабного эксперимента сервер должен располагаться на удаленной машине, а клиент на Вашей рабочей машине. Если у Вас есть доступ к удаленному серверу (VPS, Облако, ...) – можете использовать его. Можно также использовать бесплатные кредиты Yandex Cloud либо другие сервисы.
3. При желании можно повторить эксперимент, когда и сервер, и клиент расположены в Облаке (можно для машин сервера и клиента выбрать разные географические зоны).
4. Вы можете также воспользоваться удаленной машиной (сервером) нашего курса, но машины будут доступны только во время занятия. Внимание:
  - a. Используйте сервер только для выполнения домашнего задания.
  - b. Используйте предоставленный Вам логин и пароль. Не используйте логин другого студента.
  - c. Пожалуйста, не запускайте на сервере IntelliJ IDEA либо другие среды либо WireShark, чтобы не занимать ресурсы.
  - d. Ваше приложение-сервер не является требовательным к ресурсам, поэтому заранее подготовьте jar файл (командой `java -jar ...`) с Вашим приложением-

сервером, скопируйте его на виртуальную машину и запускайте именно jar файл из командной строки (не из IntelliJ IDEA).

- e. Выделяйте Вашему приложению-серверу не более 50-100 МБ оперативной памяти: use the arguments `-Xms<memory> -Xmx<memory>`. Use M or G after the numbers for indicating Megs and Gigs of bytes respectively. `-Xms` indicates the minimum and `-Xmx` the maximum.
- f. Используйте для своего сервера порт, указанный в инструкциях по подключению к удаленному серверу (колонок PORT FOR HW1); указанные в инструкции порты были предварительно открыты специально для целей выполнения Вашей домашней работы, другие порты могут быть закрыты извне.
- g. Ваше клиентское приложение должно размещаться на Вашей локальной машине.

Если Вы работаете на Windows, то для доступа к виртуальным машинам на Linux можно использовать инструменты WinSCP и Putty, а к виртуальным машинам на Windows (в том числе серверу нашего курса) – удаленный рабочий стол.

**Оценивание данного задания будет происходить следующим образом.** Во-первых, код будет анализироваться на точность выполнения требований этого задания. Далее, будет выполнен анализ полученных графиков. Затем будет учитываться качество кода (форматирование, обработка исключений, иные стандартные элементы качества кода). Дополнительно может быть проведена устная защита выборки домашних работ. Наконец, Ваша оценка будет дополнена тестом (будет проходить на ближайшем семинаре по материалам лекций и семинаров). Вклад данного задания в оценку HW1 – до 80%, вклад теста – до 20%.