



**VII Поволжская открытая олимпиада по информационным технологиям
«Волга ИТ – 2014»
Номинация «Интернет-программирование (Java)»
Задание финального этапа**

1. Робот, собирающий информацию об уникальных доменах

Вы все, конечно, знаете как работает поисковый робот (Yandex, Google и так далее).

Очень приближенно, поисковая система состоит из двух частей - робота, который постоянно обходит интернет в поисках новых сайтов и робота, который потом скачивает содержимое этих найденных страниц и индексирует их.

Робот, который находит новые страницы работает довольно просто - он стартует с одной страницы в интернете, собирает все ссылки что есть на этой странице и "проходит" по этим ссылкам на новые страницы. С этих новых страниц робот опять собирает все ссылки и повторяет процесс. Таким образом робот прыгает со страницы на страницу, с домена на домен и обходит весь Интернет.

Мы предлагаем вам написать подобного робота, но немного упростить его - собирать информацию только про домены.

Пример использования:

- 1) Запускаем робота, в качестве параметра указываем ему какую-либо страницу. Например эту - <http://stackoverflow.com/questions/5834808/designing-a-web-crawler>
- 2) Робот начинает обход Интернета и сообщает обо всех найденных уникальных доменах в консоль
- 3) Чем больше робот успеет найти уникальных доменов за 2 минуты - тем лучше

2. Утилита для синхронизации содержимого большого файла

На компьютере **A** есть один большой файл, до нескольких десятков гигабайт размером. Этот файл периодически меняется, причем меняется по разному - иногда меняется несколько байт в произвольном месте, а иногда практически весь файл.

На компьютере **B** есть копия этого файла (или пока еще нет, если синхронизация еще не запускалась). Мы периодически запускаем на компьютере **B** приложение для синхронизации и получаем последнее актуальное состояние файла.

Вам предлагается создать клиент-серверное приложение, решающее эту задачу.

Пример использования:

- 1) На машине **A** запускается сервер, в качестве параметра ему передается путь к файлу
- 2) На машине **B** запускается клиент, в качестве параметра ему передается адрес сервера и каталог, куда необходимо сохранять файл
- 3) Клиент синхронизирует содержимое файла и выходит
- 4) Мы вносим изменения в файл
- 5) Запускаем клиента еще раз, клиент опять делает синхронизацию и завершает свою работу
- 6) Обратите внимание что сервер все это время работает
- 7) Опционально - сервер одновременно позволяет осуществлять синхронизацию нескольким клиентам

Учитывая что файл может меняться очень незначительно, было бы неправильно копировать его целиком каждый раз. Подумайте над возможностью передавать только информацию об измененных частях файла.

Использовать можно любые Java-фреймворки и технологии.

И клиент и сервер будут запускаться с параметрами "-Xmx300m -Dfile.encoding=UTF-8"

3. **Файловый клиент/сервер**

Есть задача – с одного компьютера просматривать содержимое файловой системы другого компьютера.

Вам предлагается создать клиент-серверное приложение, решающее эту задачу.

Пример использования:

- 1) Запускаю на одном компьютере сервер, который ожидает подключений на любом заранее выбранном порту
- 2) На другом компьютере запускаю клиент, который подключается к серверу и отображает содержимое файловой системы сервера (директории, файлы и так далее)
- 3) Клиент может перемещаться по файловой системе (переход в выбранную папку, переход на уровень выше)
- 4) Интерфейс - GUI или консоль, по желанию
- 4) Опционально - возможность скачать выбранный файл с сервера на компьютер клиента

Использовать можно любые Java-фреймворки и технологии.

И клиент и сервер будут запускаться с параметрами "-Xmx300m -Dfile.encoding=UTF-8"