

Задание 01

1. Разработайте приложение **09-01**, представляющее собой WebSocket (WS)–сервер, прослушивающий порт **4000**.
2. WS-сервер предназначен для приема по ws-каналу файлов.
3. Принятый по ws-каналу файл записывается в директорию **upload**.
4. Разработайте приложение **09-01a**, представляющее собой WS-клиент, демонстрирующий работоспособность сервера.

Задание 02

5. Разработайте приложение **09-02**, представляющее собой WebSocket (WS)–сервер, прослушивающий порт **4000**.
6. WS-сервер предназначен для отправки по ws-каналу файлов из директории **download**.
7. Разработайте приложение **09-02a**, представляющее собой WS-клиент, демонстрирующий работоспособность сервера.

Задание 03

8. Разработайте приложение **09-03**, представляющее собой WebSocket (WS)–сервер, прослушивающий порт **4000**.
9. Сервер каждые 15 секунд всем подключившимся клиентам высылает сообщение следующего формата:

09-03-server: n, где **n** – последовательный номер отправляемого сервером сообщения.

10. С помощью **ping/pong**-механизма сервер проверяет работоспособность соединений, каждые 5 секунд, при этом сервер выводит на консоль количество работоспособных соединений.
11. Разработайте приложение **09-03a**, представляющее собой WS-клиент, демонстрирующий работоспособность сервера. Продемонстрируйте работу сервера с несколькими экземплярами **09-03a**.

Задание 04

12. Разработайте приложение **09-04**, представляющее собой WebSocket (WS)-сервер, прослушивающий порт **4000**.
13. Сервер принимает сообщение вида:

{client:**x**, timestamp:**t**}, где **x**-имя клиента, а **t**-штамп времени.

Сообщение передается клиентом в json-формате.

14. Сервер отправляет в ответ клиенту сообщение вида:

{server: **n** client:**x**, timestamp:**t**}, где **n** -номер сообщения, **x**-имя клиента, а **t**-штамп времени.

Сообщение передается сервером в json-формате.

15. Разработайте приложение **09-04a**, представляющее собой WS-клиент, демонстрирующий работоспособность сервера. Приложение принимает параметр командной строки, значение которого используется в качестве значения **x**, в сообщении для сервера.
16. Продемонстрируйте взаимодействие сервера с несколькими клиентами (клиенты должны иметь разные значения параметра).

Задание 05

17. Разработайте приложение **09-05**, представляющее собой WebSocket (WS)-сервер, прослушивающий порт **4000**.
18. Сервер обеспечивает следующий RPC-интерфейс:

RPC метод	public protected	Описание RPC-метода
square	public	если принимает один параметр r , то возвращается площадь круга радиуса r ; если принимает два параметра a и b , то возвращается площадь прямоугольника с длинами сторон a и b ;
sum	public	принимает переменное количество числовых параметров, возвращает сумму значений всех параметров;

mul	public	принимает переменное количество числовых параметров, возвращает произведение значений всех параметров;
fib	protected	принимает один числовой параметр n , возвращает массив, содержащий n элементов последовательности Фибоначчи;
fact	protected	принимает один числовой параметр n , возвращает факториал числа n ;

19. Разработайте приложение **09-05a**, представляющее собой WS-клиент, демонстрирующий работоспособность сервера. Приложение осуществляет следующие RPC-вызовы:

```
square(3), square(5,4),
sum(2), sum(2,4,6,8,10),
mul(3), mul(3,5,7,9,11,13),
fib(1), fib(2), fib(7),
fact(0), fact(5), fact(10)
```

20. Результаты вычислений отобразите в консоли приложения.
21. Разработайте приложение **09-05b**, представляющее собой WS-клиент, демонстрирующий работоспособность сервера. Приложение осуществляет параллельные (async/parallel) RPC-вызовы из п.20. Результаты вычислений отобразите в консоли приложения.
22. Разработайте приложение **09-05c**, представляющее собой WS-клиент, демонстрирующий работоспособность сервера. Приложение вычисляет с помощью RPC-вызовов следующее выражение:

```
sum(square(3), square(5,4), mul(3,5,7,9,11,13))
+fib(7)
*mul(2,4,6)
```

Результаты вычислений отобразите в консоли приложения.

Задание 06

23. Разработайте приложение **09-06**, представляющее собой WebSocket (WS)-сервер, прослушивающий порт **4000**.
24. Приложение может генерировать три события: **A**, **B**, **C**.

25. Генерация событий осуществляется при получении соответствующего сообщения через стандартный поток ввода (через консоль). При вводе символа **А** сервер генерирует событие **А**; при вводе символа **В** сервер генерирует событие **В**; при вводе символа **С** сервер генерирует событие **С**.
26. Разработайте приложение **09-06а**, представляющее собой WS-клиент, демонстрирующий работоспособность сервера. Приложение подписывается на событие **А** и сообщает о наступлении этого события выводом на консоль.
27. Разработайте приложение **09-06б**, представляющее собой WS-клиент, демонстрирующий работоспособность сервера. Приложение подписывается на событие **В** и сообщает о наступлении этого события выводом на консоль.
28. Разработайте приложение **09-06с**, представляющее собой WS-клиент, демонстрирующий работоспособность сервера. Приложение подписывается на событие **С** и сообщает о наступлении этого события выводом на консоль.
29. Продемонстрируйте совместную работу всех четырех приложений.

Задание 07

30. Разработайте приложение **09-07**, представляющее собой WebSocket (WS)-сервер, прослушивающий порт **4000**.
31. Приложение может принимать три типа уведомлений: **А**, **В**, **С**. При получении уведомления, сервер выводит соответствующее сообщение на консоль.
32. Разработайте приложение **09-07а**, представляющее собой WS-клиент, демонстрирующий работоспособность сервера. Приложение шлет уведомления серверу при получении соответствующего сообщения через стандартный поток ввода (консоль).

Задание 08. Ответьте на следующие вопросы.

33. Поясните понятие «TCP-порт».
34. Поясните понятие «сетевой сокет».
35. Поясните понятие «WebSocket».
36. Поясните процедуру установки соединения между WS-сервером и WS-клиентом.
37. Поясните понятие «широковещательное сообщение».

38. Поясните принцип организации потокового ввода/вывода через WS-соединение.
39. Поясните принцип действия и назначение механизма «ping/pong».
40. Поясните аббревиатуру «RPC».
41. Поясните принцип работы RPC-механизма, предоставляемого пакетом rpc-websockets.
42. Поясните принцип действия и назначение механизма «subscriber/publisher».
43. Поясните принцип действия и назначение механизма уведомлений.