

Задание 01

1. Разработайте приложение **11-01**, представляющий собой WebSocket (WS)-сервер, прослушивающий порт **4000**.
2. WS-сервер предназначен для приема по ws-каналу файлов.
3. Принятый по ws-каналу файл переписывается в директорию **upload**.
4. Разработайте приложение **11-01a**, представляющий собой WS-клиент, демонстрирующий работоспособность сервера.

Задание 02

5. Разработайте приложение **11-02**, представляющий собой WebSocket (WS)-сервер, прослушивающий порт **4000**.
6. WS-сервер предназначен для отправки по ws-каналу файлов из директория **download**.
7. Разработайте приложение **11-02a**, представляющий собой WS-клиент, демонстрирующий работоспособность сервера.

Задание 03

8. Разработайте приложение **11-03**, представляющий собой WebSocket (WS)-сервер, прослушивающий порт **4000**.
9. Сервер всем подключившимся клиентам каждые 15 секунд высылает сообщение следующего формата **11-03-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 в консоль количество работоспособных соединений.
11. Разработайте приложение **11-03a**, представляющий собой WS-клиент, демонстрирующий работоспособность сервера. Продемонстрируйте работу сервера с несколькими экземплярами **11-03a**.

Задание 04

12. Разработайте приложение **11-04**, представляющий 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 **11-04a**, представляющий 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 **11-05**, представляющий собой WebSocket (WS)-сервер, прослушивающий порт **4000**.
18. Сервер обеспечивает 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 **11-05a**, представляющий 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 **11-05b**, представляющий собой WS-клиент, демонстрирующий работоспособность сервера. Приложение осуществляет параллельный (async/parallel) RPC-вызовы из п.20. Результаты вычислений отобразите в консоли приложения.
22. Разработайте приложение **11-05c**, представляющий 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 **11-06**, представляющий 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 **A**, сервер генерирует событие **A**; при вводе символа **B**, сервер генерирует событие **B**; при вводе символа **C**, сервер генерирует событие **C**.

26. Разработайте приложение **11-06a**, представляющий собой WS-клиент, демонстрирующий работоспособность сервера. Приложение подписывается на событие **A** и сообщает о наступлении этого события выводом на консоль.
27. Разработайте приложение **11-06b**, представляющий собой WS-клиент, демонстрирующий работоспособность сервера. Приложение подписывается на событие **B** и сообщает о наступлении этого события выводом на консоль.
28. Разработайте приложение **11-06c**, представляющий собой WS-клиент, демонстрирующий работоспособность сервера. Приложение подписывается на событие **C** и сообщает о наступлении этого события выводом на консоль.
29. Продемонстрируйте совместную работу всех четырех приложений.

Задание 07

30. Разработайте приложение **11-07**, представляющий собой WebSocket (WS)-сервер, прослушивающий порт **4000**.
31. Приложение может принимать три типа уведомлений: **A**, **B**, **C**. При получении уведомления, сервер выводит соответствующее сообщение на консоль.
32. Разработайте приложение **11-07a**, представляющий собой WS-клиент, демонстрирующий работоспособность сервера. Приложение шлет уведомления серверу при получении соответствующего сообщения через стандартный ввод (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.