

## Вопросы к лабораторной работе №4

### 1. Дайте пояснению понятию «событие программного объекта».

Событие в программном объекте – это процесс перехода объекта из одного состояния в другое. При этом, об этом переходе могут быть извещены другие объекты. У события есть издатель (или генератор) события и могут быть подписчики (или обработчики) события.

### 2. Какой встроенный механизм используется в Node.js для генерации и обработки событий. Поясните принцип его работы.

В Node.js для генерации и обработки событий используется встроенный модуль EventEmitter. Принцип его работы основан на паттерне "Издатель-Подписчик" (Publisher-Subscriber). Этот модуль предоставляет разработчикам инструменты для создания и управления собственными объектами, которые могут генерировать события и иметь множество функций-обработчиков (листенеров), подписанных на эти события.

Вот основные шаги работы с модулем EventEmitter в Node.js:

#### 1. Подключение модуля EventEmitter:

Для начала, вы должны импортировать или подключить модуль EventEmitter из библиотеки Node.js. Это можно сделать следующим образом:

```
const EventEmitter = require('events');
```

#### 2. Создание экземпляра объекта EventEmitter:

После подключения модуля, вы создаете экземпляр объекта EventEmitter, который будет служить источником событий. Например:

```
const myEmitter = new EventEmitter();
```

#### 3. Генерация событий:

Источник событий (в данном случае, `myEmitter`) может генерировать события с использованием метода `emit()`. Вызов этого метода и указание имени события активизирует все связанные с ним функции-обработчики. Например:

```
myEmitter.emit('myEvent', 'Сообщение данных');
```

#### 4. Добавление обработчиков событий:

Вы можете добавить функции-обработчики (листенеры) к объекту `EventEmitter` с помощью метода ``on()'` или ``addListener()'`. Эти функции будут вызваны при генерации события. Например:

```
myEmitter.on('myEvent', (data) => {  
  console.log('Событие произошло с данными:', data);  
});
```

#### 5. Обработка событий:

Когда событие будет сгенерировано, все зарегистрированные обработчики будут вызваны с переданными данными. В данном примере, когда событие `'myEvent'` генерируется, вызывается функция-обработчик, и сообщение с данными выводится на консоль.

Этот механизм позволяет легко создавать асинхронные приложения в `Node.js` и обеспечивать обработку событий, такие как обработка запросов веб-сервера, управление файловой системой, взаимодействие с базой данных и многое другое.