Частное учреждение образования

«Колледж бизнеса и права»

|  |
| --- |
| УТВЕРЖДАЮ  Заведующий  методическим кабинетом  \_\_\_\_\_\_\_\_\_\_Е.В.Паскал  «\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_ |

|  |  |
| --- | --- |
| Специальность:  2-40 01 01 «Программное обеспечение информационных технологий» | Дисциплина: «Скриптовые языки программирования» |

**Лабораторная работа № 18**

**Инструкционно-технологическая карта**

Тема: Способы обработки даты и времени в JavaScript.

Цель: Научиться использовать дату и время в JavaScript, использовать и обрабатывать дату и время

Время выполнения: 2 часа

# Порядок выполнения работы

1. Изучить теоретические сведения к лабораторной работе.
2. Реализовать алгоритм решения задачи.
3. Отлаженную, работающую программу сдать преподавателю. Работу программы показать с помощью самостоятельно разработанных тестов.
4. Ответить на контрольные вопросы.

# Теоретические сведения

Новый встроенный объект: **Date**. Он содержит дату и время, а также предоставляет методы управления ими.

Например, его можно использовать для хранения времени создания/изменения, для измерения времени или просто для вывода текущей даты.

## Создание

Для создания нового объекта **Date** нужно вызвать конструктор **new Date()** с одним из следующих аргументов:

#### new Date()

Без аргументов – создать объект **Date** с текущими датой и временем:

let now = new Date();

alert( now ); // показывает текущие дату и время

#### new Date(milliseconds)

Создать объект **Date** с временем, равным количеству миллисекунд (тысячная доля секунды), прошедших с 1 января 1970 года UTC+0.

// 0 соответствует 01.01.1970 UTC+0

let Jan01\_1970 = new Date(0);

alert( Jan01\_1970 );

// теперь добавим 24 часа и получим 02.01.1970 UTC+0

let Jan02\_1970 = new Date(24 \* 3600 \* 1000);

alert( Jan02\_1970 );

Целое число, представляющее собой количество миллисекунд, прошедших с начала 1970 года, называется таймстамп (англ. timestamp).

Это – легковесное численное представление даты. Из таймстампа всегда можно получить дату с помощью **new Date(timestamp)** и преобразовать существующий объект **Date** в таймстамп, используя метод **date.getTime()** (см. ниже).

Датам до 1 января 1970 будут соответствовать отрицательные таймстампы, например:

// 31 декабря 1969 года

let Dec31\_1969 = new Date(-24 \* 3600 \* 1000);

alert( Dec31\_1969 );

#### new Date(datestring)

Если аргумент всего один, и это строка, то из неё «прочитывается» дата. Алгоритм разбора – такой же, как в **Date.parse**, который мы рассмотрим позже.

let date = new Date("2017-01-26");

alert(date);

// Время не указано, поэтому оно ставится в полночь по Гринвичу и

// меняется в соответствии с часовым поясом места выполнения кода

// Так что в результате можно получить

// Thu Jan 26 2017 11:00:00 GMT+1100 (восточно-австралийское время)

// или

// Wed Jan 25 2017 16:00:00 GMT-0800 (тихоокеанское время)

#### new Date(year, month, date, hours, minutes, seconds, ms)

Создать объект Date с заданными компонентами в местном часовом поясе. Обязательны только первые два аргумента.

* **year** должен состоять из четырёх цифр: значение 2013 корректно, 98 – нет.
* **month** начинается с **0** (январь) по **11** (декабрь).
* Параметр **date** здесь представляет собой день месяца. Если параметр не задан, то принимается значение **1**.
* Если параметры **hours/minutes/seconds/ms** отсутствуют, их значением становится **0**.

Например:

new Date(2011, 0, 1, 0, 0, 0, 0); // // 1 Jan 2011, 00:00:00

new Date(2011, 0, 1); // то же самое, так как часы и проч. равны 0

Максимальная точность – 1 мс (до 1/1000 секунды):

let date = new Date(2011, 0, 1, 2, 3, 4, 567);

alert( date ); // 1.01.2011, 02:03:04.567

## Получение компонентов даты

Существуют методы получения года, месяца и т.д. из объекта Date:

**getFullYear()** – получить год (4 цифры)

Никакого **getYear().** Только **getFullYear()**. Многие интерпретаторы JavaScript реализуют нестандартный и устаревший метод **getYear(),** который порой возвращает год в виде двух цифр. Его стоит обходить стороной. Если нужно значение года, используйте **getFullYear()**.

**getMonth()** – получить месяц, от 0 до 11.

**getDate()** – получить день месяца, от 1 до 31, что несколько противоречит названию метода.

**getHours(), getMinutes(), getSeconds(), getMilliseconds()** – получить, соответственно, часы, минуты, секунды или миллисекунды.

**getDay()** – вернуть день недели от 0 (воскресенье) до 6 (суббота). Несмотря на то, что в ряде стран за первый день недели принят понедельник, в JavaScript начало недели приходится на воскресенье.

**Все вышеперечисленные методы возвращают значения в соответствии с местным часовым поясом.**

Однако существуют и их UTC-варианты, возвращающие день, месяц, год для временной зоны UTC+0: **getUTCFullYear**(), **getUTCMonth**(), **getUTCDay**(). Для их использования требуется после «**get**» подставить «**UTC**».

Если местный часовой пояс смещён относительно UTC, то следующий код покажет разные часы:

// текущая дата

let date = new Date();

// час в вашем текущем часовом поясе

alert( date.getHours() );

// час в часовом поясе UTC+0 (лондонское время без перехода на летнее время)

alert( date.getUTCHours() );

Помимо вышеприведённых методов, существуют два особых метода без UTC-варианта:

**getTime**() – для заданной даты возвращает таймстамп – количество миллисекунд, прошедших с 1 января 1970 года UTC+0.

**getTimezoneOffset**() – возвращает разницу в минутах между UTC и местным часовым поясом:

// если вы в часовом поясе UTC-1, то выводится 60

// если вы в часовом поясе UTC+3, выводится -180

alert( new Date().getTimezoneOffset() );

## Установка компонентов даты

Следующие методы позволяют установить компоненты даты и времени:

* setFullYear(year, [month], [date])
* setMonth(month, [date])
* setDate(date)
* setHours(hour, [min], [sec], [ms])
* setMinutes(min, [sec], [ms])
* setSeconds(sec, [ms])
* setMilliseconds(ms)
* setTime(milliseconds) (устанавливает дату в виде целого количества миллисекунд, прошедших с 01.01.1970 UTC)

У всех этих методов, кроме **setTime**(), есть UTC-вариант, например: **setUTCHours**().

Из примера выше видно, что некоторые методы могут устанавливать сразу несколько компонентов даты, например: **setHours**. Если какая-то компонента не указана, она не меняется.

Пример:

let today = new Date();

today.setHours(0);

alert(today); // выводится сегодняшняя дата, но значение часа будет 0

today.setHours(0, 0, 0, 0);

alert(today); // всё ещё выводится сегодняшняя дата, но время будет ровно 00:00:00.

## Автоисправление даты

Автоисправление – это очень полезная особенность объектов Date. Можно устанавливать компоненты даты вне обычного диапазона значений, а объект сам себя исправит.

Пример:

let date = new Date(2013, 0, 32); // 32 Jan 2013 ?!?

alert(date); // ...1st Feb 2013!

Неправильные компоненты даты автоматически распределяются по остальным.

Например, требуется увеличить дату «28 февраля 2016» на два дня. В зависимости от того, високосный это год или нет, результатом будет «2 марта» или «1 марта». Нам об этом думать не нужно. Просто прибавляем два дня. Объект **Date** позаботится об остальном:

let date = new Date(2024, 1, 28);

date.setDate(date.getDate() + 2);

alert( date ); // 1 Mar 2016

Эту возможность часто используют, чтобы получить дату по прошествии заданного отрезка времени. Например, получим дату «спустя 70 секунд с текущего момента»:

let date = new Date();

date.setSeconds(date.getSeconds() + 70);

alert( date ); // выводит правильную дату

Также можно установить нулевые или даже отрицательные значения. Например:

let date = new Date(2016, 0, 2); // 2 Jan 2016

date.setDate(1); // задать первое число месяца

alert( date );

date.setDate(0); // первый день месяца – это 1, так что выводится последнее число предыдущего месяца

alert( date ); // 31 Dec 2015

## Преобразование к числу, разность дат

Если объект Date преобразовать в число, то получим таймстамп по аналогии с **date.getTime():**

let date = new Date();

alert(+date); // количество миллисекунд, то же самое, что date.getTime()

Важный побочный эффект: даты можно вычитать, в результате получаем разность в миллисекундах.

Этот приём можно использовать для измерения времени:

let start = new Date(); // начинаем отсчёт времени

// выполняем некоторые действия

for (let i = 0; i < 100000; i++) {

  let doSomething = i \* i \* i;

}

let end = new Date(); // заканчиваем отсчёт времени

alert( `Цикл отработал за ${end - start} миллисекунд` );

## Date.now()

Если нужно просто измерить время, объект **Date** нам не нужен.

Существует особый метод **Date.now**(), возвращающий текущую метку времени.

Семантически он эквивалентен **new Date().getTime**(), однако метод не создаёт промежуточный объект **Date**. Так что этот способ работает быстрее и не нагружает сборщик мусора.

Данный метод используется из соображений удобства или когда важно быстродействие, например, при разработке игр на JavaScript или других специализированных приложений.

Вероятно, предыдущий пример лучше переписать так:

let start = Date.now(); // количество миллисекунд с 1 января 1970 года

// выполняем некоторые действия

for (let i = 0; i < 100000; i++) {

  let doSomething = i \* i \* i;

}

let end = Date.now(); // заканчиваем отсчёт времени

alert( `Цикл отработал за ${end - start} миллисекунд` ); // вычитаются числа, а не даты

## Бенчмаркинг

Стоит помнить, если нужно точно протестировать производительность функции, которая зависит от процессора.

Например, сравним две функции, вычисляющие разницу между двумя датами: какая сработает быстрее?

Подобные вычисления, замеряющие производительность, также называют «бенчмарками» (benchmark).

// есть date1 и date2, какая функция быстрее вернёт разницу между ними в миллисекундах?

function diffSubtract(date1, date2) {

    return date2 - date1;

}

// или

function diffGetTime(date1, date2) {

    return date2.getTime() - date1.getTime();

}

Обе функции делают буквально одно и то же, только одна использует явный метод **date.getTime**() для получения даты в миллисекундах, а другая полагается на преобразование даты в число. Результат их работы всегда один и тот же.

Но какая функция быстрее?

Для начала можно запустить их много раз подряд и засечь разницу. В нашем случае функции очень простые, так что потребуется хотя бы 100000 повторений.

Проведём измерения:

function diffSubtract(date1, date2) {

    return date2 - date1;

}

function diffGetTime(date1, date2) {

    return date2.getTime() - date1.getTime();

}

function bench(f) {

    let date1 = new Date(0);

    let date2 = new Date();

    let start = Date.now();

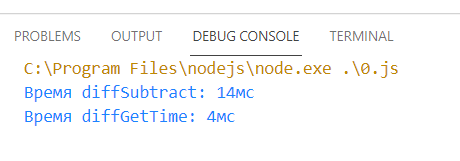
    for (let i = 0; i < 100000; i++) f(date1, date2);

    return Date.now() - start;

}

console.log('Время diffSubtract: ' + bench(diffSubtract) + 'мс');

console.log('Время diffGetTime: ' + bench(diffGetTime) + 'мс');



Метод **getTime**() работает ощутимо быстрее! Всё потому, что не производится преобразование типов, и интерпретаторам такое намного легче оптимизировать.

## Разбор строки с датой

Метод **Date.parse(str)** считывает дату из строки.

Формат строки должен быть следующим: **YYYY-MM-DDTHH:mm:ss.sssZ**, где:

* **YYYY-MM-DD** – это дата: год-месяц-день.
* Символ **«T»** используется в качестве разделителя.
* **HH:mm:ss.sss** – время: часы, минуты, секунды и миллисекунды.
* Необязательная часть **«Z»** обозначает часовой пояс в формате **+-hh:mm.** Если указать просто букву Z, то получим UTC+0.
* Возможны и более короткие варианты, например, **YYYY-MM-DD** или **YYYY-MM**, или даже **YYYY**.

Вызов **Date.parse(str)** обрабатывает строку в заданном формате и возвращает таймстамп (количество миллисекунд с 1 января 1970 года UTC+0). Если формат неправильный, возвращается **NaN**.

Например:

let ms = Date.parse('2012-01-26T13:51:50.417-07:00');

alert(ms); // 1327611110417 (таймстамп)

Можно тут же создать объект **new Date** из таймстампа:

let date = new Date( Date.parse('2012-01-26T13:51:50.417-07:00') );

alert(date);

# Контрольные вопросы

1. Что такое таймстамп?
2. В чём измеряется таймстамп? Какой дате соответствует таймстамп со значением 0?
3. От какого дня отсчитываются дни недели в функции getDay()? Какому дню соответствует getDay() равный 0?
4. Какому месяцу соответствует месяц с номером 0?

# Задания

1. Создать дату. Создать объект Date для даты: 20 февраля 2012 года, 3 часа 12 минут. Временная зона – местная.
2. Показать день недели. Написать функцию getWeekDay(date), показывающую день недели в коротком формате: «ПН», «ВТ», «СР», «ЧТ», «ПТ», «СБ», «ВС».
3. День недели в европейской нумерации. В Европейских странах неделя начинается с понедельника (день номер 1), затем идёт вторник (номер 2) и так до воскресенья (номер 7). Написать функцию getLocalDay(date), которая возвращает «европейский» день недели для даты date.
4. Какой день месяца был много дней назад? Создать функцию getDateAgo(date, days), возвращающую число, которое было days дней назад от даты date. К примеру, если сегодня двадцатое число, то getDateAgo(new Date(), 1) вернёт девятнадцатое и getDateAgo(new Date(), 2) – восемнадцатое. Функция должна надёжно работать при значении days=365 и больших значениях.
5. Написать функцию formatDate(date), форматирующую date по следующему принципу:

* Если спустя date прошло менее 1 секунды, вывести «прямо сейчас».
* В противном случае, если с date прошло меньше 1 минуты, вывести «n сек. Назад».
* В противном случае, если меньше часа, вывести «m мин. Назад».
* В противном случае, полная дата в формате «DD.MM.YY HH:mm». А именно: «день.месяц.год часы:минуты», всё в виде двух цифр, т.е. 31.12.16 10:00.

Например:

alert( formatDate(new Date(new Date - 1)) ); // "прямо сейчас"

alert( formatDate(new Date(new Date - 30 \* 1000)) ); // "30 сек. назад"

alert( formatDate(new Date(new Date - 5 \* 60 \* 1000)) ); // "5 мин. назад"

// вчерашняя дата

alert( formatDate(new Date(new Date - 86400 \* 1000)) );

1. Выполнить задание по вариантам.

|  |  |
| --- | --- |
| № | Задание |
|  | 1. Ввести две даты и определить количество лет между ними 2. Написать программу расчета стоимости потребленной электроэнергии в трёхзоновом электросчетчике. Исходные параметры: время, количество потребленной электроэнергии в кВт и полная стоимость 1 кВт/час (в копейках). Расчет суммы к оплате производить с учетом процентного отношения стоимости тарифов для временных зон: 1-я зона времени 8:00 – 11:00 и 20:00 – 23:00 оплата 100%, 2-я зона времени 7:00 – 8:00 и 11:00 – 20:00 оплата 70%, 3-я зона времени 23:00 – 7:00 оплата 40%. |
|  | 1. Ввести две даты и вывести сообщение с информацией, какая из двух введенных дат более поздняя. 2. Написать программу контроля электропитания ноутбука. Исходные параметры: время включения ноутбука от батареи и задекларированное производителем время работы от батареи в часах без подключения к сети электропитания. Программа информирует о количестве часов работы ноутбука от батареи на текущий момент времени и сколько осталось минут до полного разряда батареи. При расчете считать, что в момент включения ноутбука батарея имела 100% заряд и способна обеспечить задекларированное производителем время работы. |
|  | 1. Вывести сообщение с информацией о текущей системной дате и времени. 2. Разработать программу для оператора мобильной связи, позволяющую для отдельно введенных значений даты и времени последнего исходящего звонка определить дату и время истечения срока действия данного пакета и какой это будет день недели (1 – понедельник). В условии договора сказано, что пакет действует в течение 1 года после последнего исходящего звонка. |
|  | 1. Ввести значение времени и выполнить его форматирование: часы (0-23), минуты (0-59) и секунды (0-59). 2. Судебному отделу разработать программу, которая для введенной даты административного правонарушения выводит информацию о сроке ее давности на данный момент времени. Срок давности представить в виде отдельных значений: полного количества дней, полного количества месяцев и полного количества лет, прошедших с момента правонарушения. |
|  | 1. Ввести четыре отдельных значения (часы, минуты, секунды и миллисекунды) и объединить их в одно значение данных типа время. 2. 2. Написать программу любителям предсказаний, позволяющую для введенной даты определить, принадлежит ли она к високосному году, если это не так, то выдать информацию о количестве лет до ближайшего високосного года. |
|  | 1. Ввести значение и проверить является ли оно данными типа дата. 2. Написать программу для медицинского персонала, выполняющую расчет времени проведения процедур. В программе реализовать ввод значений: время первой процедуры, время следующей процедуры и общее количество назначенных процедур. Программа должна вывести список времени, по которому следует выполнять назначенные процедуры. В расчетах считать, что время между всеми процедурами равное. |
|  | 1. Вывести сообщение с информацией о текущем системном времени. 2. Оператору телефонной сети написать программу, которая для введен- ных значений времени начала звонка, времени окончания звонка и стои- мости одной минуты разговора (в копейках), выдает информацию о длительности звонка в секундах и сумме денег (в гривнах), снятых со счета клиента. |
|  | 1. Ввести дату и разбить ее на три отдельные составляющие: день, месяц и год. 2. Сети супермаркетов разработать программу для контроля срока годности продуктов питания. В качестве исходных значений потребитель вводит дату и время производства продукта, а также срок годности в часах. Результатом работы программы должно быть информационное сообщение о количестве оставшихся часов на данный момент времени до истечения срока годности продукта, а в случае его истечения ‑ предупреждение об этом. |
|  | 1. Вывести сообщение с информацией о текущей системной дате. 2. Написать для телевизионной компании программу, которая будет выполнять расчет: времени окончания телепередачи, ее продолжительность в минутах и количество в ней рекламных пауз. Исходными значениями программы будут: введенное время начала выхода в эфир телепередачи, ее продолжительность в секундах, а также условие договора, что вставка рекламной паузы производится каждые 5 минут эфирного времени. |
|  | 1. Ввести значение и проверить, является ли оно данными типа время. 2. Разработать систему, отложения начала выполнения программы посудомоечной машинки на указанное количество часов. Систему представить в виде программы, в которой пользователь вводит длительность работы посудомоечной машинки в минутах и время отсрочки начала запуска программы в часах. Определить дату и время начала мойки, а также дату и время ее окончания относительно текущих значений даты и времени в момент нажатия кнопки «Пуск». |
|  | 1. Ввести дату и определить номер дня недели, к которому она принадлежит (1 – понедельник). 2. Написать программу электронного календаря, которая для введенной даты рождения выдаст сообщение о количестве оставшихся дней до дня рождения в этом году, а в случае, если день рождения уже прошел, то количество дней, прошедших после него. |
|  | 1. Ввести два значения времени и определить количество секунд между ними. 2. Информационному отделу железнодорожного вокзала написать программу формирования бегущей строки. Исходные значения: направление поезда (например, Одесса-Киев), время отправления, номер платформы и номер пути. Программа должна формировать текст, бегущий строки следующего вида: «Поезд следованием направление отправляется от номер\_платформы платформы номер\_пути пути через количество\_минут\_до\_отправления мин». Курсивный текст, следует заменить соответствующими значениями. Для расчета количества минут до отправления использовать введенное время отправления и текущее системное время. |
|  | 1. Ввести дату и выполнить ее форматирование: день в числовом виде (01- 31), а также в виде аббревиатуры (пн - вс), месяц с начальными нулями (01-12) и год в полном формате (0000 – 9999). 2. Отделу, занимающемуся сбором и анализом дат исторических событий, написать программу, которая для введенной даты и времени исторического события определяет, на какой минуте от начала года оно произошло, и какой это был день недели (1 – понедельник). |
|  | 1. Ввести значение времени и выполнить его форматирование: часы (00- 23), минуты (00-59) и секунды (00-59). 2. Написать программу контроля предвыборной кампании, например в Верховный совет, которая для введенной даты голосования выдает сообщение, информирующее о количестве оставшихся дней до начала голосования и дату окончания предвыборной агитации. По закону предвыборная агитация прекращается в ноль часов по местному времени за сутки до дня голосования. |
|  | 1. Ввести два значения времени и определить количество минут между ними. 2. Отделу кадров предприятия, занимающемуся планированием переноса отгула государственных праздников, припавших на выходные дни, написать программу, которая для отдельно введенных значений даты (день и месяц) государственного праздника выводит сообщение какой это день недели в этом году (1 - понедельник) и если праздник припадает на воскресенье, то печатает дату следующего за ним понедельника. |
|  | 1. Ввести две даты и определить количество месяцев между ними. 2. Написать программу управления таймером стиральной машинки. Исходные данные: введенное значение длительности стирки в минутах и текущее значение времени и даты в момент нажатия кнопки «Пуск». Определить дату и время окончания стирки. |
|  | 1. Ввести три отдельных значения (год, месяц и день) и объединить их в одно значение данных типа дата. 2. Организовать контроль времени работы программного обеспечения. Исходные параметры: текущее системное время в момент запуска программы и в момент ее закрытия. Перед закрытием программа должна выводить информацию о времени начала ее работы, времени ее окончания и длительности работы программы в миллисекундах. |
|  | 1. Ввести дату и выполнить ее форматирование: день с начальным нулем (01-31), месяц в виде полного названия (Январь – Декабрь) и год в виде двух последних цифр. 2. Реализовать часть программного комплекса «Бухгалтерский учет», которая для введенной даты выдает информацию, к какому отчетному периоду (кварталу) она принадлежит, и сколько прошло дней от начала года до введенной даты. |
|  | 1. Ввести две даты и определить количество дней между ними. 2. Диспетчерскому центру трамвайного депо требуется разработать программу, которая для введенных значений времени выхода трамвая на маршрут и времени его возврата в депо определяет общее время пребывания трамвая на маршруте в минутах и количество целых часов. |
|  | 1. Ввести дату и выполнить ее форматирование: вначале числовое значение дня без начальных нулей, затем полное название дня недели (Понедельник – Воскресенье), далее месяц в виде полного названия (Январь – Декабрь) и в конце год в полном формате (0000 – 9999). 2. Написать программу, которая для введенных значений времени старта и времени финиша легкоатлета определяет, за сколько секунд он прошел дистанцию 3000 м и его среднюю скорость в м/с. |
|  | 1. Ввести два значения времени и определить количество часов между ними. 2. Разработать дополнительный модуль для текстового редактора. Задача модуля выполнять замену месяца, введенной даты в числовом формате, на его полное название (Январь-Декабрь), а также выводить информационное сообщение о количестве дней в этом месяце. |
|  | 1. Ввести значение времени и выполнить его форматирование по 12- часовой шкале с использованием на конце символов am и pm. 2. Для клуба любителей авиации и космонавтики написать программу расчета времени до старта. Исходные параметры: дата и время предстоящего запуска космического корабля. Результат: количество полных месяцев, дней, часов, минут и секунд до начала старта на текущий момент времени. Всю информацию представить в виде отдельных значений. |
|  | 1. Ввести значение времени и определить количество минут прошедших от начала суток до введенного значения. 2. Написать программу, которая для двух введенных значений времени финиша гоночных автомобилей определяет победителя, пришедшего первым, и количество миллисекунд, обеспечивших выигрыш (разрыв во времени). |
|  | 1. Ввести значение время и разбить его на четыре составляющих: часы, минуты, секунды и миллисекунды. 2. Написать программу будильника, которая для введенного значения времени сигнала выдает информацию о количестве оставшихся минут до начала сигнала относительно текущего системного времени и на какой минуте от начала суток он начнется. |
|  | 1. Ввести значение времени и выполнить его форматирование по 12- часовой шкале с использованием на конце символов a и p. 2. Написать программу управления микроволновой печью, которая для введенного количества минут разогрева блюда определяет время завершения разогрева. В качестве начального времени разогрева использовать текущее значение системного времени в момент нажатия кнопки «Старт». |

1. Реализовать классы, имеющие следующие поля:

|  |  |  |
| --- | --- | --- |
| № | Класс и поля | Задача |
|  | Сеансы радиоэфира:   * позывной; * частота; * дата сеанса; * время начала сеанса; * время окончания; * количество переданных групп (последовательность из 5 символов). | * вывести информацию о скорости передачи групп в минуту по каждому из сеансов; * ввести дату и два значения времени, организовать просмотр информации о выходе радистов на связь для введенной даты на указанном интервале времени. |
|  | Сотрудники предприятия:   * табельный номер; * ФИО сотрудника; * дата рождения; * пол (м/ж); * дата поступления на работу; * должность; * оклад; | * вывести информацию о возрасте каждого сотрудника на данный момент времени и количестве дней, отработанных им на предприятии; * организовать просмотр информации о сотрудниках пенсионного возраста (мужчины – до 60 лет, женщины до 55 лет), проработавших на предприятии более 30 лет. |
|  | Биллинговая система интернет- провайдера:   * логин пользователя; * дата выхода в online; * время начала сеанса online; * время завершения; * объем принятой информации (Кб); * объем отправленной информации (Кб). | * вывести информацию о длительности по каждому из сеансов online и объеме потребленного трафика за сеанс в Мб (принято + отправлено); * ввести две даты и организовать для введенного интервала просмотр информации о пользователях, выходивших в online и потребивших объем трафика (принято + отправлено) за сеанс более 20 Мб. |
|  | Городское строительство:   * название строительной компании; * название объекта стройки; * количество строящихся метров (м2); * дата начала строительства; * планируемая дата окончания строительства; * состояние (строится, сдан и т.д.). | * вывести информацию по всем объектам с указанием количества дней планируемого строительства; * вывести все названия строящихся объектов, окончание стройки которых запланировано на 4-й квартал текущего года. |
|  | Счетчик посещения сайта:   * адрес открытой web-странички; * ip-адрес посетителя; * web-браузер посетителя; * дата визита; * время визита; | * вывести информацию обо всех посещениях сайта с указанием дня посещения (1 - понедельник); * ввести два значения времени и вывести сведения о посещении web-странички indexphp во введенный временной интервал прошлого дня. |
|  | Продуктовый магазин:   * наименование товара; * фирма производитель; * дата производства; * срок годности в сутках; * цена. | * вывести информацию по всем продуктам с указанием даты окончания срока годности; * ввести номер месяца и вывести информацию обо всех товарах, произведенных в этом месяце текущего года. |
|  | Расписание экзаменационной сессии:   * дата; * предмет; * ФИО преподавателя; * группа; * № аудитории; * время начала; * время окончания. | * вывести информацию по всем экзаменам с указанием дня недели (1 – понедельник); * вывести информацию по экзаменам, начало которых приходится на первую половину дня указанной даты (до 12:00). |
|  | Расписание поездов:   * № поезда; * станция отправления; * время отправления; * станция прибытия; * время прибытия. | * вывести сведения по всем поездам с указанием длительности времени пребывания в пути; * ввести два значения времени и вывести для данного интервала времени все сведения о поездах, которые отправляются из Киева. |
|  | Журнал событий ОС Windows:  название приложения;  уровень события (ошибка, предупреждение и т.д.);  код события;  дата события;  время события. | вывести информацию по всем сообщениям с указанием в днях срока их давности на текущий момент (сколько дней прошло с момента события);  ввести дату и вывести сведения об ошибках, произошедших за введенную дату. |
|  | Журнал учета посещения сотрудниками работы:   * дата; * ФИО сотрудника; * должность; * время прихода; * время ухода. | * вывести сведения по всем записям с указанием количества часов пребывания сотрудника на работе; * ввести дату и вывести информацию о сотрудниках, пришедших на работу во второй половине дня (после 12:00) указанной даты. |
|  | Афиша кинотеатра:   * название кинотеатра; * дата киносеанса; * название кинофильма; * жанр; * время начала; * длительность в минутах. | * вывести сведения о времени завершения каждого из киносеансов; * вывести информацию о фильмах, идущих в выходные дни (Суббота и Воскресенье). |
|  | Статистика отправки сообщений с почтового сервера:   * e-mail отправителя; * e-mail получателя; * дата отправки; * время отправки; * размер сообщения (Кб). | вывести сведения обо всех отправленных сообщения с указанием срока давности отправки в днях (количество дней, прошедших с момента отправки до текущего дня);  ввести дату и интервал времени (два отдельных значения времени) и вывести сведения обо всех отправителях, пославших сообщения в указанную дату и интервал времени. |
|  | Телевизионная студия:   * название канала; * планируемая дата проведения профилактики; * время начала профилактических работ; * время окончания профилактики. | * вывести информацию по всем каналам с указанием длительности профилактических работ в минутах; * вывести информацию о каналах время начала профилактики, которых запланировано на ночное время суток (с 22:00 до 6:00). |
|  | Список группы студентов:   * № пп; * ФИО студента; * день рождения; * месяц рождения; * возраст. | * вывести по каждому студенту полную дату рождения и номер дня недели, в который он родился (1 ‑ понедельник); * вывести список студентов, родившихся в високосный год. |
|  | Телефонная сеть:   * ФИО абонента; * номер телефона; * дата поломки; * время поломки; * дата устранения; * время устранения. | * вывести сведения по всем заявкам на ремонт с указанием срока устранения неисправности в днях; * вывести все сведения о поломках за прошлый месяц. |
|  | Салон мебели:   * № заказа; * дата заказ; * ФИО заказчика; * адрес заказчика; * срок выполнения в днях; * стоимость заказа. | * вывести список всех заказов с указанием даты их выполнения; * ввести номер месяца и отобразить информацию о заказах, сделанных в указанный месяц этого года. |
|  | График поставки товара на склад:   * дата; * название фирмы поставщика; * ФИО директора фирмы; * время поставки; * вес груза (кг). | * вывести предполагаемое время окончания разгрузки для каждого поставщика из расчета, что на разгрузку 100 кг затрачивается 20 минут; * вывести информацию по поставщикам, планирующим поставку товара на завтра в первой половине дня (до 12:00). |
|  | Поликлиника:   * название кабинета; * номер кабинета; * ФИО врача; * день приема (формат: 1\_Понедельник – 7\_Воскресенье); * время начала приема; * время окончания. | * вывести информацию по всем записям с указанием длительности приема; * ввести значение времени и дня недели приема; определить, принимает ли кабинет флюорографии в это время. |
|  | Средства защиты от поражения электрическим током:   * инвентарный номер; * наименование; * ФИО ответственного; * дата последней проверки; * очередность проверки в месяцах. | * вывести сведения обо всех средствах защиты с указанием для них следующей даты проверки; * вывести информацию о средствах защиты, проверка которых запланирована на следующий месяц этого года. |
|  | Система учета коммунальных платежей:   * лицевой счет; * ФИО потребителя; * услуга; * начислено к оплате; * дата оплаты; * сумма оплаты. | * вывести сведения по всем платежам с указанием номера декады, в которой был произведен платеж; * вывести сведения по платежам за последние три месяца. |
|  | Статистика сдачи экзаменов на права в ГАИ:   * дата сдачи; * ФИО сдававшего; * время начала теста; * время окончания теста; * количество правильных ответов. | * вывести информацию по всем результатам с указанием затраченного времени на сдачу теста в минутах; * вывести информацию о положительных результатах сдачи тестов в прошлом месяце из расчета, что общее количество вопросов 20 и разрешено допустить до 2 ошибок. |
|  | Украинские отопительные системы:   * населенный пункт; * № котельной; * количество отапливаемых объектов; * дата начала отопительного сезона; * температура воздуха на начало отопительного сезона; * дата окончания отопительного сезона. | * вывести сведения по всем котельным с указанием длительности отопительного сезона в днях; * вывести сведения по всем котельным, начавшим отопительный сезон после 15 октября. |
|  | Техническая поддержка интернет-провайдера:   * дата звонка; * время звонка; * адрес абонента; * характер неисправности; * состояние заявки (устранено, ведутся работы и т.д.). | * вывести информацию по всем заявкам с указанием номера дня недели звонка (1 – понедельник); * вывести сведения обо всех невыполненных заявках, поступивших за последние три дня. |
|  | Маршрут движения почтовых машин:   * № маршрута; * пункт назначения; * время выезда на маршрут; * время прибытия в назначенный пункт; * длительность времени обмена почты в минутах. | вывести информацию по всем маршрутам с указанием затраченного времени на путь (в минутах);  ввести два значение времени; вывести информацию о маршрутах, выезд которых планируется в указанный интервал времени и длительность времени обмена почты не превышает 10 минут. |

# Литература

**Диков, А.В.** Клиентские технологии веб программирования: JavaScript и DOM: учебное пособие / А.В. Диков. – СПб: Лань, 2020 – 124 с.

**Читанамбри, Кирупа.** Изучаем React / Кирупа Читанамбри – 2-е изд. – М.: Эксмо, 2019. 368 с.

**Хавербеке, Марейн.** Выразительный JavaScript. Современное веб-программирование. / Марейн Хавербеке – 3-е изд. – СПб.: Питер, 2019. – 480 с.:

Преподаватель Рогалевич А.В.

Рассмотрено на заседании цикловой

комиссии программного обеспечения

информационных технологий №10

Протокол № \_\_\_ от \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Председатель ЦК\_\_\_\_\_\_\_\_ В.Ю.Михалевич