ПрИЛОЖЕНИЕ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ИМЕНИ. В.Ф. УТКИНА

Кафедра «Вычислительная и прикладная математика»

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ по дисциплине
«Web-программирование»**

Направление подготовки

 09.03.04 «Программная инженерия»

Направленность (профиль) подготовки

Программная инженерия

Квалификация выпускника – бакалавр

Форма обучения – очная

Рязань

1. **ОБЩИЕ ПОЛОЖЕНИЯ**

Оценочные материалы – это совокупность учебно-методических материалов и процедур, предназначенных для оценки качества освоения обучающимися данной дисциплины как части основной образовательной программы.

*Цель* – оценить соответствие знаний, умений и уровня приобретенных компетенций, обучающихся целям и требованиям основной образовательной программы в ходе проведения текущего контроля и промежуточной аттестации.

*Основная задача* – обеспечить оценку уровня сформированности компетенций, приобретаемых обучающимся в соответствии с этими требованиями.

Контроль знаний обучающихся проводится в форме промежуточной аттестации – зачет.

**2 ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ**

Сформированность каждой компетенции в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

1. пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;
2. продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенций по завершении освоения дисциплины;
3. эталонный уровень характеризуется максимально возможной выраженностью компетенций и является важным качественным ориентиром для самосовершенствования.

**Уровень освоения компетенций, формируемых дисциплиной**

*а) описание критериев и шкалы оценивания тестирования:*

|  |  |
| --- | --- |
| **Шкала оценивания** | **Критерий** |
| 3 балла(эталонный уровень) | уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 85 до 100% |
| 2 балла(продвинутый уровень) | уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 75 до 84% |
| 1 балл(пороговый уровень) | уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 60 до 74% |
| 0 баллов | уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 0 до 59% |

*б) описание критериев и шкалы оценивания теоретического вопроса:*

|  |  |
| --- | --- |
| ***Шкала оценивания*** | **Критерий** |
| 3 балла(эталонный уровень) | выставляется студенту, который дал полный ответ на вопрос, показал глубокие систематизированные знания, смог привести примеры, ответил на дополнительные вопросы преподавателя. |
| 2 балла(продвинутый уровень) | выставляется студенту, который дал полный ответ на вопрос, но на некоторые дополнительные вопросы преподавателя ответил только с помощью наводящих вопросов. |
| 1 балл(пороговый уровень) | выставляется студенту, который дал неполный ответ на вопрос в билете и смог ответить на дополнительные вопросы только с помощью преподавателя.  |
| 0 баллов | выставляется студенту, который не смог ответить на вопрос  |

*в) описание критериев и шкалы оценивания практического задания:*

|  |  |
| --- | --- |
| **Шкала оценивания** | **Критерий** |
| 3 балла(эталонный уровень) | Задание решено верно |
| 2 балла(продвинутый уровень) | Задание решено верно, но имеются технические неточности в выполнении |
| 1 балл(пороговый уровень) | Задание решено верно, с дополнительными наводящими вопросами преподавателя |
| 0 баллов | Задание не решено |

На экзаменвыносится: тестовое задание, 1 практическое задание и 1 теоретический вопрос. Студент может набрать максимум 9 баллов. Итоговый суммарный балл студента, полученный при прохождении промежуточной аттестации, переводится в традиционную форму по системе «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

|  |  |
| --- | --- |
| **Шкала оценивания** | **Критерий** |
| отлично(эталонный уровень) | 8 – 9 баллов | Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий и лабораторных работ. |
| хорошо(продвинутый уровень) | 6 – 7 баллов |
| удовлетворительно(пороговый уровень) | 4 – 5 баллов |
| неудовлетворительно | 0 – 3 баллов | Студент не выполнил всех предусмотренных в течение семестра текущих заданий  |

**3 ПАСПОРТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ПО ДИСЦИПЛИНЕ**

|  |  |  |
| --- | --- | --- |
| **Контролируемые разделы (темы) дисциплины** | **Код контролируемой компетенции (или её части)** | **Наименование****оценочного****средства** |
|
| Тема 1. Гипертекстовая ИС и введение в Javascript. Появление технологии WEB. Появление языка Javascript, задачи и область его применения. | ПК-3.1 ПК-3.2 ПК-3.4 | зачет |
| Тема 2. Лексическая структура JavaScript | ПК-3.1 ПК-3.2 ПК-3.4 | зачет |
| Тема 3. Типы данных. Значения. Переменные. | ПК-3.1 ПК-3.2 ПК-3.4 | зачет |
| Тема 4. Операции и операторы. | ПК-3.1 ПК-3.2 ПК-3.4 | зачет |
| Тема 5. Массивы. Методы класса Array. | ПК-3.1 ПК-3.2 ПК-3.4 | зачет |
| Тема 6. Объекты и их свойства. | ПК-3.1 ПК-3.2 ПК-3.4 | зачет |
| Тема 7. Сериализация и клонирование объектов. | ПК-3.1 ПК-3.2 ПК-3.4 | зачет |
| Тема 8. Функции. Базовое использование функций в языке Javascript. | ПК-3.1 ПК-3.2 ПК-3.4 | зачет |
| Тема 9. Расширенное использование функций в языке Javascript. | ПК-3.1 ПК-3.2 ПК-3.4 | зачет |
| Тема 10. Классы объектов в языке Javascript. | ПК-3.1 ПК-3.2 ПК-3.4 | зачет |
| Тема 11. Применение функций-конструкторов в языке Javascript. | ПК-3.1 ПК-3.2 ПК-3.4 | зачет |

Для заочной формы обучения дополнительно предусмотрены контрольные работы в 1 и 2 семестре, включающие все контролируемые разделы (темы) дисциплины.

**4 ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ**

**4.1 Промежуточная аттестация (экзамен)**

|  |
| --- |
| **ПК-3: Способен выполнять работы и управление работами по созданию и сопровождению информационных систем** |
| **ПК-3.1. Разрабатывает, анализирует и утверждает требования к информационной системе** |
|  |
| **ПК-3.2. Выполняет проектирование и реализацию информационной системы** |
| **ПК-3.4. Выполняет развертывание информационной системы у заказчика** |

***а) типовые тестовые вопросы закрытого типа***

1. HTML это:

 Язык гипертекстовой информационной системы

 **Язык гипертекстовой разметки документов**

 Язык гипертекстовой передачи данных

 Язык гипертекстовой связи браузерных приложений

2. URL это:

 **Универсальный способ адресации ресурсов**

 Универсальное хранилище данных

 Универсальный протокол передачи данных

 Универсальный гипертекстовый язык

3. HTTP это:

 Протокол обмена графической информацией

 Протокол обмена служебной информацией

 **Протокол обмена гипертекстовой информацией**

 Протокол обмена данными по безопасному соединению клиентов

4. Три технологии, которые должен знать любой frontend разработчик:

 HTTP, PHP, Javascript

 HTML, CGI, Javascript

 Linux, CSS, Javascript

 **HTML, CSS, Javascript**

5. Схема доступа URL это:

 Стандартная программа, визуализации web-страниц

 Стандартная программа, связывающая клиент и сервер.

 **Стандартная программа, которой передаётся адрес URL для обработки.**

 Стандартная программа, которая запрашивает файлы с сервера.

6. Какими символами задаётся многострочный комментарий?

 // … //

 //\* … \*//

"""… """

 **/\* … \*/**

7. На какие две группы по структурному представлению можно разделить все типы данных в JS?

 **Простые типы и объекты**

 Классы и объекты

 Простые типы и литералы

 Простые типы и константы

8. На какие две группы по модификации значений можно разделить все типы данных в JS?

 **Мутабельные и иммутабельные**

 Гомогенные и гетерогенные

 структурные и однородные

 вещественные и целые

9. Какие типы данных есть в JS для представления чисел?

 Только вещественные 34-бит

 целые 64-бит и вещественные 64-бит

 **Только вещественные 64-бит**

 целые 32-бит и вещественные 32-бит

10. Какие значения в результате автоматического приведения типов дают значение false?

 undefined, [false], 0, -0, NaN, ""

 **undefined, null, 0, -0, NaN, ""**

 undefined, null, 0, -0, {}, ""

 undefined, null, 0, -0, NaN, "false"

11. Какой идентификатор допустим в JS:

 **$var**

 12var

 @var

 var

12. Какое действие допустимо в JS:

 const a = 5; a = 6;

 const a = [1, 2, 3]; a = [4, 5, 6];

 **const a = [1, 2, 3]; a[1] = 6;**

 const a = 5; a = [];

13. Какой метод объекта String **НЕ** возвращает подстроку?

 substr

 **split**

 slice

 substring

14. Второй аргумент метода replace объекта String задаёт:

 строку, на которую происходит замена

 строку, которую ищем для замены

 регулярное выражение, определяющее замену

 **строку, на которую происходит замена или функция, возвращаемую строку замены**

15. Оператор switch сравнивает значение селектора со значениями констант через оператор:

==

 isEqual()

 **=== (правильный ответ)**

 =

16. Какого объекта BOM **НЕ** существует?

 location

 screen

 XMLHttpRequest

 **browser**

17. Как в модели DOM обратиться к следующему соседнему узлу?

 parentNode

 **nextSibling**

 nextChild

 nextNode

18. Как найти все элементы DOM по заданному CSS селектору?

 **elem.querySelectorAll(css)**

 elem.getSelectorAll(css)

 elem.queryAllSelectors(css)

 elem.recieveSelectorAll(css)

19. Получить значение атрибута элемента можно через метод:

 elem.attribute(name)

 elem.returnAttribute(name)

 **elem.getAttribute(name)**

 elem.hasAttribute(name)

20. Добавить дочерний узел можно через метод:

 parent.addChild(elem)

 parent.insertChild(elem)

 parent.pushChild(elem)

 **parent.appendChild(elem)**

***б) типовые тестовые вопросы открытого типа***

1. Что такое гипертекст? (Гипертекст - текст, представленный в виде ассоциативно связанных автономных блоков. Позволяет включать в страницы ссылки на другие части данного или другого документа)

2. Что такое гипертекстовая информационная система? (Гипертекстовая информационная система – это система, состоящая из множества информационных узлов, множества гипертекстовых связей, определенных на этих узлах и инструментах манипулирования узлами и связями.)

3. Что такое ECMAScript? (это стандарт языка, которому следуют все реализации Javascript)

4. Какие есть три способа функционального применения Javascript на веб-страницах? (1. Схема URL – для выполнения JS кода была введена своя схема доступа – Javascript. 2. Обработчики событий – код JS, указываются в атрибутах контейнеров, с которыми эти события связаны. 3) Вставка через контейнер SCRIPT — это возможность генерации текста документа JavaScript-кодом.)

5. Какие две функции выполняет контейнер Script? (1. Размещение кода внутри HTML-документа 2. Условная генерация HTML-разметки на стороне браузера)

6. Дайте лексическую характеристику языка Javascript (используется набор символов Юникода, чувствительный к регистру символов, для отделения инструкций друг от друга в языке JavaScript используется точка с запятой ;)

7. Что такое объект в JS? (Объект т.е. член объектного типа данных представляет собой коллекцию свойств, каждое из которых имеет имя и значение (либо простого, либо объектного). Обычный объект JavaScript представляет собой неупорядоченную коллекцию именованных значений.)

8. Что такое функция в JS? (функция это объект, с которым связан выполняемый код. Функция может вызыватьсядля выполнения определенной операции и возвращать вычисленное значение. Функции являются настоящими значениями, поэтому программы JavaScript могут манипулировать ими, как обычными объектами.)

9. Какие есть виды числовых литералов в JS? (десятичные 6545, шестнадцатеричные 0xff, вещественные числа 5.45, вещественные в экспоненциальной нотации 6.2e23)

10. Чем характеризуется строковый тип в JS? (Строка – это неизменяемая упорядоченная последовательность 16-битных значений, каждое из которых представляет символ Юникода. Строки в JavaScript являются типом данных, используемым для представления текста.)

11. В чем отличие значений null и undefined? (undefined можно рассматривать как признак неожиданного или ошибочного отсутствия какого-либо значения, а null – как признак обычного или вполне ожидаемого отсутствия.)

12. Что такое сборка мусора в JS? (программист может создавать объекты в памяти, но ему нет необходимости беспокоиться об уничтожении этих объектов и освобождении занимаемой ими памяти. Когда объект выходит за пределы области видимости (т. е. когда программа утрачивает возможность доступа к этому объекту) и интерпретатор обнаруживает, что данный объект никогда больше не сможет использоваться, он автоматически освобождает занимаемую им память.)

13. В чем отличие в объявлении переменных через var и let? (Для переменных объявленных через var Javascript блок не задает область видимости. Поэтому был введен специальное ключевое слово let, позволяющее объявить переменную для конкретного блока программы.)

14. Какие есть специальные числовые значения в JS? (Infinity, -Infinity, 0, -0, NaN)

15. В чем разница между операторами равенства == и тождества ===? (оператор равенства == выполняет преобразование типов перед сравнением, а оператор === тождества сразу возвращает false при несовпадении типов операндов)

16. Что такое DOM? (DOM (document object model) – Глобальный объект document даёт возможность взаимодействовать с содержимым страницы. Согласно DOM‑модели, документ является иерархическим представлением совокупности тегов. Каждый HTML‑тег образует узел дерева, а вложенные в него теги становятся дочерними узлами. )

17. Что такое BOM? (BOM (browser object model) - вспомогательные объекты, для управления функциями браузера, кроме непосредственно документа.)

18. Как можно найти элементы DOM по названию тега? (Вызов elem.getElementsByTagName(tag) ищет все элементы с заданным тегом tag внутри элемента elem и возвращает их в виде коллекции. Можно получить всех потомков, передав звездочку '\*')

19. В чем отличие свойств innerHTML и outerHTML? (innerHTML - внутреннее содержимое узла‑элемента в виде HTML. Можно изменять. outerHTML - полный HTML узла‑элемента вместе с тегом. Можно изменять, но в переменной сохраниться старый элемент.)

20. Зачем нужны методы работы с атрибутами в DOM элементах? (При генерации DOM‑модели большинство стандартных HTML‑атрибутов становятся свойствами соответствующих объектов. Но бывают ситуации, когда значение атрибутов и свойств различается или аналогичное свойство отсутствует. Т.к. узел DOM – это объект JavaScript, поэтому он может содержать также и пользовательские свойства, и методы, которые будут видны только в JavaScript и никак не будут влиять на отображение документа. Для доступа к таким атрибутам как раз и нужны данные методы)

21. Напишите пример создания двумерного массива 10x10 в JS

 var table = new Array(10); // в таблице 10 строк
 for(let col = 0; col < table.length; ++col) table[col] = new Array(10);

22. Напишите пример обхода массива через оператор for in:
 var arr = [1, 2, 3, 4, 5];
 for(let i in arr) console.log("arr[" + i + "] = " + arr[i]);

23. Напишите пример функции, возвращающей уникальное целое число при каждом своем вызове

 function uniqueID()
 {
 // инициализировать свойство id если оно не было добавлено до этого
 if(uniqueID.id === undefined)
 uniqueID.id = 0;
 return uniqueID.id++; // увеличить и вернуть id
 }

24. Напишите функцию факториала, которая сохраняет результаты предыдущих вычислений в своём собственном свойстве

 function fact(n)
 {
 // валидация параметра
 n = n < 1 ? 1 : n;
 if(fact.cache === undefined)
 fact.cache = [, 1]; // инициализируем кэш базовым случаем
 if(!(n in fact.cache)) // если значение не сохранялось ранее
 fact.cache[n] = n \* fact(n - 1); // вычислить и сохранить
 return fact.cache[n]; // вернуть сохранёный в кеше результат
 }

25. Напишите пример функции, создающую независимую копию объекта путем глубокого копирования (все свойства объекта являются либо приметивными типами, либо тоже пользоватлескими объектами):

 function makeDeepCopy(obj)
 {
 var copy = {};
 for(var key in obj)
 {
 if(typeof obj[key] != "object")
 copy[key] = obj[key];
 else
 copy[key] = makeDeepCopy(obj[key]);
 }
 return copy;
 }

**4.2 Типовые контрольные вопросы к экзамену**

1) Гипертекст. Гипертекстовая информационная система.

2) Технология World Wide Web. URL, HTTP, HTML.

3) Язык JavaScript. Основные характеристики.

4) Основные способы функционального применения JavaScript. Примеры.

5) Лексическая структура JavaScript. Регистр букв. Пробелы, переводы строк

и символы управления форматом. Комментарии. Примеры.

6) Лексическая структура JavaScript. Литералы. Идентификаторы. Точки с запятой. Примеры.

7) Типы данных. Значения. Переменные.

8) Сборка мусора. Изменяемые и неизменяемые типы данных. Числа. Целые литералы.

Примеры.

9) Литералы вещественных чисел. Дата и время. Логические значения. Строки. Примеры.

10) Значения null и undefined. Объявление переменных. Область видимости переменной.

Примеры.

11) Операции и операторы. Приоритет выполнения. Назначение. Примеры.

12) Объект Math. Основные методы. Работа со строками. Примеры.

13) Специальные значения. Преобразования типов. Преобразование и равенство. Явные

преобразования. Примеры.

14) Оператор if. Оператор switch. Операторы while, do…while. Оператор for. Примеры.

15) Коллекция. Массивы и их характеристики. Создание массивов. Примеры.

16) Чтение, запись, добавление и удаление элементов массива. Обход элементов массива.

Примеры.

17) Многомерные массивы. Методы класса Array. Метод join. Метод reverse. Метод concat.

Примеры.

18) Методы класса Array. Метод slice. Метод splice. Методы push() и pop(), unshift() и shift() для

работы с массивом как со стеком. Примеры.

19) Объекты. Свойства объектов. Создание объектов. Получение, изменение, удаление

свойств объекта. Примеры.

20) Объекты. Свойства объектов. Создание объектов. Проверка существования свойств.

Перечисление свойств. Примеры.

21) Функции. Параметры и аргументы. Возвращаемое значение. Контекст вызова функций.

Определение функций. Примеры.

22) Вложенные функции. Вызов функций. Вызов функций как методов. Необязательные

аргументы. Примеры.

23) Списки аргументов функций переменной длины. Определение собственных свойств

функций. Функции как пространства имён. Примеры.

24) Замыкания. Методы call и apply. Примеры.

25) Конструктор Function. Его особенности. Функции высшего порядка. Меморизация функций.

Примеры.

26) Классы объектов. Экземпляры. Прототипы. Примеры.

27) Наследование. Функция inherit. Классы и прототипы. Фабричная функция. Примеры.

**4.3 Типовые задачи к экзамену по дисциплине**

1. Составить функцию нахождения среднего арифметического элементов вектора L. Используя данную функцию, найти средние арифметические в строках матриц М1 и М2 и сформировать из них соответственно вектора V1, V2. Написать функцию сортировки векторов V1 и V2 по возрастанию.

2. Составить функцию проверки упорядоченности символьных элементов вектора L по алфавиту. Используя данную функцию, проанализировать строки символьных матриц M1 и М2. Для строк, которые не отсортированы по алфавиту, выделить их элементы в отдельный вектор и написать функцию, удаляющую все буквы из них, с нечётным номером в алфавите.

3. Составить функцию, подсчитывающую количество слов вектора, которые начинаются и оканчиваются одной и той же буквой. Используя данную функцию, найти суммы числа слов в каждой строке в матрицах M1 и М2 и записать их все в один суммарный вектор V. Найти с помощью отдельной функции минимальный и максимальный элемент вектора V и удалить эти элементы из него.

4. Составить функцию, которая перемещает в начало вектора L все четные элементы, а в конец вектора - нечетные элементы. С использованием данной функции преобразовать все строки матриц M1 и М2. Далее написать функцию нахождения максимального и минимального элемента в матрицах М1 и М2, а после этого удалить данные элементы из них.

5. Составить функцию, проверяющую на равенство значения элементов векторов L1 и L2 и возвращающую вектор одинаковых элементов, присутствующих в обоих из них. Используя функцию, проанализировать соответствующие строки матриц М1 и М2, и из полученных векторов одинаковых элементов создать матрицу M3, где все несуществующие элементы заменены значением «бесконечность». Написать функцию, подсчитывающую кол-во значений «бесконечность» в матрице M3.

6. Составить функцию, определяющую включения вектора L1 в вектор L2 и наоборот (варианты возвращаемого значения: 0 – вектор L1 включает L2, 1 – вектор L2 включает L1, 2 – вектора полностью идентичны, 3 – вектора не включают друг друга). Используя функцию, проанализировать соответствующие строки матриц М1 и М2, составить суммарный вектор из всех не включающих друг друга строк. Написать функцию, замещающую в данном векторе все дублирующийся элементы значением «бесконечность».

7. Составить функцию вставки элемента Е после каждого элемента вектора, превышающего некоторое значение Р. Используя данную функцию обработать все строки матриц M1 и М2. Определить на сколько данные матрицы увеличились, уменьшились или остались прежнего размера. После написать функцию, находящую количество элементов Е с хотя бы одним чётным индексом в матрицах М1 и М2.

8. В векторе натуральных чисел переставить элементы по следующему правилу: если текущий элемент больше некоторого числа Р, то поместить следующий за ним элемент в конец вектора; если текущий элемент меньше или равен числу Р, перенести в начало вектора текущий элемент (первый оставить без изменения). Используя данную функцию обработать все строки матриц M1 и М2. Далее написать функцию, которая удаляет из матриц М1 и М2 строки с большим количеством отрицательных чисел, чем положительных. Определить на сколько данные матрицы увеличились, уменьшились или остались прежнего размера.

9. Написать функцию создания вектора L1 из вектора L2, расположив в нём только положительные элементы вектора L2 в обратном порядке. С помощью данной функции обработать строки матриц М1 и М2. Написать функцию обработки матриц, которая замещает все чётные элементы значением «бесконечность» и применить её к матрицам М1 и М2.

10. Написать функцию определяющую, входит ли элемент Е в вектор L, подсчитать количество вхождений данного элемента в вектор и вставить первый элемент вектора после каждого вхождения Е. С помощью данной функции обработать строки матриц М1 и М2. После этого написать функцию удаления элемента из матрицы и с помощью неё удалить из матриц М1 и М2 элемент Е. Определить на сколько данные матрицы увеличились, уменьшились или остались прежнего размера, относительно обработанных ранее матриц.

11. Написать функцию возвращающую вектор L, упорядочив его по убыванию, из четных элементов вектора L1 и нечетных элементов вектора L2. С помощью данной функции сформировать матрицу М, где каждая строка матрицы формируется из элементов строк матрицы M1 и строк матрицы М2. После написать функцию, заполняющую отсутствующие элементы матрицы М минимальным элементом данной строки в чётных строках, и максимальным элементом в нечётных строках.

12. Написать функцию, формирующую два вектора L1 и L2 из вектора L по следующему правилу: в вектор L1 занести положительные элементы, а в вектор L2 – отрицательные. С помощью данной функции сформировать построчно матрицы М1 и М2 из строк матрицы М. После написать функцию, находящую вектор минимальных элементов каждой строки матрицы М1 и максимальных элементов строк матрицы М2.

13. Описать массив объектов и поместить в него сгенерированные сведения о N книгах. Предусмотреть такие сведения как название книги, жанр, дата издания (отдельный объект), количество экземпляров, ФИО автора (отдельным объектом), количество страниц. Написать функцию выдачи списка книг по фамилии автора, жанру или диапазону годов издания. Написать функцию удаления сведений о количестве страниц, если количество страниц менее заданного числа. Написать функцию добавления информации о возрасте книги, найденую по дате её издания.

14. Описать массив объектов и поместить в него сгенерированные сведения о N работниках. Предусмотреть такие сведения как ФИО работника (отдельным объектом), дата рождения (отдельный объект), номер телефона, место работы (отдельный объект со сведениями о названии организации, должности и стаже). Написать функцию выдачи списка работников по названию организации, должности или диапазону стажа. Написать функцию удаления сведений о дате рождения, если стаж менее заданного числа. Написать функцию добавления информации о районе проживания работника, найденного по первым двум цифрам телефона.

15. Описать массив объектов и поместить в него сгенерированные сведения о N студентах. Предусмотреть такие сведения как ФИО студента (отдельным объектом), дата поступления (отдельный объект), номер телефона, результаты сессии (отдельный массив объектов с информацией о названии предметов и полученных оценках). Написать функцию удаления сведений о дате поступления, если год поступления старше заданного. Написать функцию добавления информации о среднем балле студента, найденного по оценкам сессии. Написать функцию выдачи списка студентов отсортированному по убыванию среднего балла.

16. Описать массив объектов и поместить в него сгенерированные сведения о N деталях. Предусмотреть такие сведения как наименование детали, габаритные размеры (отдельный объект), материал, масса детали, список поставщиков деталей (отдельный массив объектов из названий организации и контактного телефона). Написать функцию удаления сведений о материале, если масса детали менее указанной величины. Написать функцию добавления информации о габаритном объёме детали, найденного по габаритным размерам. Написать функцию выдачи списка деталей отсортированному по убыванию массы.

17. Описать массив объектов и поместить в него сгенерированные сведения о N сданных экзаменационных сессий. Предусмотреть такие сведения сессии как номер курса, дата начала сессии (отдельный объект), дата конца сессии (отдельный объект), список предметов (массив объектов со сведениями о названии предмета и полученной оценки). Написать функцию удаления сведений о номере курса, если номер сессии нечётный. Написать функцию добавления информации о средней оценки сессии, найденного по списку оценок предметов. Написать функцию выдачи списка предметов и оценок лучшей сессии и худшей сессии.

18. Описать массив объектов и поместить в него сгенерированные сведения о N людях. Предусмотреть такие сведения как ФИО человека (отдельным объектом), пол, дата рождения (отдельный объект), номер телефона, адрес проживания (отдельный объект содержащий сведения о городе, улице, номере дома и номере квартиры). Написать функцию выдачи списка людей по городу, полу или диапазону годов рождения. Написать функцию удаления сведений о дате рождения, если год рождения более указанного. Написать функцию добавления информации о районе проживания работника, найденного по первым двум цифрам телефона.

19. Описать массив объектов и поместить в него сгенерированные сведения о N работниках. Предусмотреть такие сведения как ФИО работника (отдельным объектом), дата рождения (отдельный объект), номер цеха, трудовая информация (отдельный объект со сведениями о должности, разряде, стаже). Написать функцию выдачи списка работников по должности, разряду или диапазону стажа. Написать функцию удаления сведений о дате рождения, если стаж менее заданного числа. Написать функцию добавления информации о возрасте работника, найденного по году рождения.

20. Описать массив объектов и поместить в него сгенерированные сведения о N сотрудниках. Предусмотреть такие сведения как ФИО сотрудника (отдельным объектом), дата рождения (отдельный объект), должность, стаж, зарплата (отдельный объект со сведениями о окладе, премии, оплате интенсивности, оплате переработки). Написать функцию удаления сведений о дате рождения, если стаж менее заданного числа. Написать функцию добавления информации о суммарном доходе работника, найденного как сумма всей составляющей зарплаты минус 13%. Написать функцию выдачи списка работников, отсортированных по убыванию дохода.

21. Описать массив объектов и поместить в него сгенерированные сведения о плане выпуска N наименований. Предусмотреть такие сведения как название изделия, шифр, единица измерения, план выпуска (отдельный объект из плана выпуска и сколько фактически выпущено), список заказчиков (отдельный массив объектов из названий организации и количества закупаемого наименования). Написать функцию удаления сведений о единице измерения, если план выпуска менее заданного числа. Написать функцию добавления информации о проценте выполнения плана, найденного как соотношение фактического выпуска от плана выпуска. Написать функцию выдачи списка изделий, с перевыполнением плана, списка изделий с недовыполнением плана.

22. Описать массив объектов и поместить в него сгенерированные сведения о N спортсменах. Предусмотреть такие сведения как ФИО спортсмена (отдельным объектом), дата рождения (отдельный объект), страна, вид соревнования, результаты соревнований (отдельный объект со сведениями о названии соревнования, дате проведения, результате спортсмена). Написать функцию выдачи списка спортсменов по названию соревнования, стране или диапазону годов рождения. Написать функцию удаления сведений о дате рождения, если год рождения менее заданного числа. Написать функцию добавления информации о среднем результате спортсмена по всем соревнованиям.

23. Описать массив объектов и поместить в него сгенерированные сведения о N футболистах. Предусмотреть такие сведения как ФИО футболиста (отдельным объектом), дата рождения (отдельный объект), количество голов, команда (отдельный объект со сведениями о названии команды, стране, дате вступления (отдельный объект), зарплате футболиста). Написать функцию выдачи списка футболистов по названию команды, стране или диапазону забитых голов. Написать функцию удаления сведений о дате рождения, если год рождения менее заданного числа. Написать функцию добавления информации о количестве лет нахождения в команде, рассчитанной по году вступления в команду.

24. Описать массив объектов и поместить в него сгенерированные сведения о инвентаризационной ведомости из N наименований. Предусмотреть такие сведения как название наименования, инвентарный номер, дата принятия на учёт (отдельный объект), количество, место хранения (отдельный объект из полей номер корпуса, номер этажа, номер помещения). Написать функцию удаления сведений о дате принятия на учёт, если год принятия является текущим. Написать функцию добавления информации о сроке службы наименования по текущей дате и дате принятия его на учёт. Написать функцию выдачи списка наименований по номеру корпуса, номеру этажа или с указанным диапазоном сроков службы.

25. «Комплексное число» – Complex. Разработать объект комплексных чисел. Структура должна работать с функциями для изменения и получения значения действительной и мнимой части, для реализации операций сложения, вычитания, умножения, деления, присваивания комплексных чисел. Предусмотреть функцию toString. Создать 2 массива структур и с помощью них поэлементно показать работу всех операций.

26. «Дробь» – Fraction. Разработать объект в виде пары целых положительных чисел (m,n) а также отдельно знак дроби. Структура должна работать с функциями для изменения и получения значения числителя и знаменателя, сложения, вычитания, умножения, деления и присваивания дробей. Предусмотреть функцию toString. Создать 2 массива структур и с помощью них поэлементно показать работу всех операций.

27. «Вектор» – Vector. Разработать объект вектора размерности n. Реализовать функции для изменения и получения значения компонента вектора, вычисления длины вектора, скалярного произведения, сложения, умножения, умножения на скаляр. Предусмотреть функцию toString. Создать 2 массива структур и с помощью них поэлементно показать работу всех операций.

28. «Квадратная матрица» – Matrix. Разработать объект квадратной матрицы n x n. Реализовать функции для изменения и получения значения элемента матрицы, сложения, вычитания, умножения матриц; вычисления индексов максимального и минимального элемента матрицы. Предусмотреть функцию toString. Создать 2 массива структур и с помощью них поэлементно показать работу всех операций.

29. «Многочлен» – Polynom. Разработать объект полинома степени n. Реализовать функции для изменения и получения значения указанного коэффициента, вычисления значения полинома; сложения, вычитания, умножения полиномов. Предусмотреть функцию toString. Создать 2 массива структур и с помощью них поэлементно показать работу всех операций.

30. «Фигуры» – Shapes. Разработать объект для описания плоских фигур: круг, прямоугольник, треугольник. Включить функции для получения и изменения параметров фигур, перемещения на плоскости, вращения, нахождения площади и периметра фигуры. Предусмотреть функцию toString. Выполнить тестирование модуля, создав массив объектов и показав на его примере работу всех функций.

31. «Множество целых чисел» –Set. Разработать объект множества целых чисел мощности n. Реализовать функции для определения принадлежности заданного элемента множеству, добавление\удаление элемента, пересечения, объединения, разности двух множеств. Предусмотреть функцию toString. Создать 2 массива структур и с помощью них поэлементно показать работу всех операций.

32. «Массив строк» – StringArray. Разработать объект для представления массива строк. Реализовать функции для добавления\удаления строк, для поэлементной конкатенации двух массивов, упорядочения строк по длине, слияния двух массивов строк с удалением повторяющихся строк, а также формирование массива количества слов в каждой строке. Предусмотреть функцию toString. Создать 2 структуры и с помощью них поэлементно показать работу всех функций.

33. «Массив бит» – BitArray. Разработать объект представляющий собой массив битов длины n. Реализовать функции для установки и получения значения бита на заданной позиции, изменения размера массива (справа и слева), сдвиг битов вправо\влево на заданное число позиций, битовые операции and и or для двух массивов. Предусмотреть функцию toString. Создать 2 массива структур и с помощью них поэлементно показать работу всех операций.

34. «Булева матрица» – BoolMatrix. Разработать объект представляющий собой матрицу булевых значений размерности n×m. Реализовать функции для изменения и получения значения указанного элемента, логического сложения, умножения и инверсии матриц. Реализовать функцию для подсчета количества true и false значений в матрице. Предусмотреть функцию toString. Создать 2 массива структур и с помощью них поэлементно показать работу всех операций.

35. «Односвязный список» – LinkedList. Разработать объект для работы с односвязным списком с целыми числами. Реализовать функции добавления элемента на заданную позицию, удаление всех элементов с заданным значением, получение значения по заданному индексу, объединение двух списков, разделение списка на два с указанной позиции, реверс списка. Предусмотреть функцию toString. Создать 2 массива структур и с помощью них поэлементно показать работу всех операций.

36. «Бинарное дерево» – BinaryTree. Разработать объект для работы с бинарным деревом, узлы которого содержат натуральные числа. Реализовать функции добавления и удаления узлов, получения массива узлов с заданным значением, определения высоты и количества листьев у дерева. Предусмотреть функцию toString. Создать массив объектов и с помощью них поэлементно показать работу всех операций.