# Лабораторная работа №7

### Задание на лабораторную работу

В процессе написания тестовых заданий ознакомиться с элементами языка XML и средствами Java для работы с XML-документами, а также с созданием html-страницы с применением CSS.

### Задание 1

Изучить предлагаемый файл описания типа документа сформировать документ согласно этим правилам.

---student.dtd ---------------------------------------------------

<?xml version="1.0" encoding="UTF-8" ?>

<!ELEMENT student (subject\*,average?)>

<!ATTLIST student

lastname CDATA #REQUIRED

>

<!ELEMENT subject EMPTY>

<!ATTLIST subject

title CDATA #REQUIRED

mark (1|2|3|4|5) #REQUIRED

>

<!ELEMENT average (#PCDATA)>

------------------------------------------------------------------

### Задание 2

Разработать на Java консольное приложение, имеющее два входных параметра: имена входного и выходного файла. Задача приложения заключается в проверке значения средней оценки и его коррекции, если в исходном документе оно не соответствует действительности. Использовать DOM - анализатор.

### Задание 3

Разработать html-страницы для ввода данных и для вывода результата.

Первая страница должна называться index.html. На ней должны находиться:

* заголовок,
* сопроводительный текст,
* два поля для ввода данных для выполнения операции,
* приглашения (подсказки) к этим полям ввода,
* радиокнопки с четырьмя арифметическими операциями,
* кнопка для завершения ввода (с переходом на result.html).

Вторая страница должна называться result.html. На ней должны находиться:

* заголовок,
* значения аргументов и результат выполнения операции, оформленные в виде таблицы,
* ваша любимая фотография,
* ссылка на исходную страницу, меняющая цвет при наведении на неё курсора.

Обе страницы должны быть оформлены с применением CSS. Таблица стилей (общая для обеих страниц) и код на JavaScript должны быть размещены в отдельных файлах. Все элементы обеих страниц должны изменить свой внешний вид в соответствии со стилями (дизайн – на ваше усмотрение).

### Вопросы

1. XML. Отличия XML от HTML. Структура XML – документа. Пример XML. Создание XML-документов. Пример.
2. DTD и XML-схема. Примеры.
3. Dom и SAX. Примеры.
4. JavaScript. Области применения. Размещение в HTML-документе. Обработка браузером. Комментарии. Литералы и переменные. Управление порядком выполнения. Примеры.
5. JavaScript. Функции. Объекты. Функции-конструкторы. Встроенные объекты. Наследование. Выбрасывание и обработка исключений.Пользовательские массивы. Методы пользовательских массивов.
6. JavaScript. Виды объектов. BOM и DOM. Объект window. Создание и закрытие новых окон. Запуск новых потоков. Свойства window(объекты **location, history, navigator, opener**). Объект document. Прямая запись в документ(document.write(значение)). События объектов (**document.getElementById("picture")**.**onчто-нибудь**). Работа с формами. Программирование гиперссылок.

**Объект window**

• Глобальный объект.

• Все объявляемые переменные и объекты становятся его свойствами.

• Содержит либо напрямую информацию о документе, либо ассоциированный массив (карт, map) фреймов (frames).

• Имеет свои свойства, методы и события.

**Объект document**

• В нём и живёт DOM.

• Это тоже ассоциированный массив специфического вида.

• Все теги получают в соответствие объект.

 Имя объекта определяется значением атрибута name тега.

 Атрибуты тегов становятся свойствами соответствующих объектов.

 Не все свойства объектов соответствуют атрибутам тегов.

• Объекты формируются по ходу чтения документа браузером.

• Некоторые виды объектов объединяются в дополнительные массивы (формы, ссылки и т.д.).

• У объектов могут быть события.

• У объектов могут быть дополнительные методы.

**Работа с формами**

* Доступ к форме:
  + По имени как к свойству документа
  + В массиве **document.forms**
* Доступ к элементам:
  + По имени как к свойству формы
  + В массиве **elements**формы
* Атрибуты тегов элементов формы и самой формы – свойства соответствующих объектов.
* События формы:
  + **onsubmit**
  + **onreset**
* Методы формы
  + **submit()**
  + **reset()**

**Программирование гиперссылок**

• Доступ к гиперссылкам:

 По имени/идентификатору

 В массиве document.links[]

 Объект имеет «тип» URL

• Объект ссылки имеет свойства и события

• Ссылка может иметь программный вид

 javascript:код

 результат работы кода может быть показан в браузере

 Результатом считается результат последнего выражения

 Приём void(0);

 Так можно не только в ссылке, но и в action у форм