Министерство сельского хозяйства Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Пермский государственный аграрно-технологический университет

имени академика Д.Н. Прянишникова»

Кафедра Информационных технологий

и программной инженерии

КУРСОВОЙ ПРОЕКТ

по дисциплине: Программная инженерия

на тему: Разработка продукционной экспертной системы.

Выполнил:

студент 3-го курса очного отделения

специальности 09.03.03 Прикладная информатика

шифр ПИб – 661 – 2019

Гагарина Алена Юрьевна

Проверил:

доцент кафедры ИТиПИ, к.т.н., доцент

Беляков Андрей Юрьевич

Пермь – 2022

Содержание

[1. Постановка задачи 3](#_Toc93684279)

[2. Разработка структуры для хранения данных 4](#_Toc93684280)

[3. Разработка приложения 6](#_Toc93684281)

[4. Заключение 9](#_Toc93684282)

[Список литературы 10](#_Toc93684283)

[Приложения 11](#_Toc93684284)

# Постановка задачи

Прежде всего, поставим задачу, для решения которой будет разрабатываться экспертная система. Подходящей задачей, при решении которой можно использовать обратную цепочку рассуждений, может быть задача, вытекающая из следующей ситуации: пользователю, задали задачу, угадать героя из определенного сериала, при помощи наводящих вопросов. Но пользователь не смотрел данный сериал, или пользователь забыл, как зовут определенного героя, и хочет вспомнить, но помнит только роль героя и внешний вид. Тогда на помощь придет экспертная система, в которой есть ряд вопросов отвечая на которые он придет к правильному ответу.

Таким образом, необходимо разработать экспертную систему, которая определит заданного героя пользователя.



Рисунок 1. Начальная форма ЭС



Рисунок 2 Форма с результатом ЭС

# Разработка структуры для хранения данных

JSON - простой, основанный на использовании текста, способ хранить и передавать структурированные данные. С помощью простого синтаксиса можно легко хранить все, что угодно, начиная от одного числа до строк, массивов и объектов, в простом тексте. Также можно связывать между собой массивы и объекты, создавая сложные структуры данных.

После создания строки JSON, ее легко отправить другому приложению или в другое место сети, так как она представляет собой простой текст.

Структура строки JSON практически ничем не отличается от записи JavaScript объекта.

Она состоит из набора пар ключ-значения. В этой паре ключ отделяется от значения с помощью знака двоеточия (:), а одна пара от другой - с помощью запятой (,). При этом ключ в JSON, в отличие от объекта обязательно должен быть заключен в двойные кавычки. Это главное отличие JSON от JavaScript объекта.

*Листинг 1. Пример кода JSON*

    "title": "ЭС может угадать загаданного героя из сериала Ривердейл",

    "start": "Роль героя",

    "dict": {

        "Роль героя": ["Главная", "Второстепенная"],

        "Главная": ["школьник", "родитель"],

        "школьник": ["мужской пол", "женский пол"],

        "мужской пол": ["блондин", "рыжий", "брюнет"],

        "брюнет": ["Джагхед Джонс"],

        "рыжий": ["Арчибальд Эндрюс"],

Дерево решений экспертной системы представлено на Рисунке 3.



Рисунок 3. Дерево решений ЭС

# Разработка приложения

Интерфейс приложения разработан с помощью языка разметки гипертекста HTML и каскадных таблиц CSS.

HTML задаёт структуру содержимого и его смысл, определяя такой контент как, к примеру, заголовки, абзацы или изображения.

HTML не задает конкретные и точные атрибуты форматирования документа. Конкретный вид документа окончательно определяет только программа-браузер на компьютере пользователя Интернета. HTML также не является языком программирования, но web-страницы могут включать в себя встроенные программы-скрипты на языках Javascript и Visual Basic Script и программы-апплеты на языке Java.

CSS — это язык презентаций, созданный для оформления внешнего вида контента, использующий, например, шрифты или цвета.

Обычно CSS-стили используются для создания и изменения стиля элементов веб-страниц и пользовательских интерфейсов, написанных на языках HTML и XHTML

Эти два языка — HTML и CSS независимы друг от друга и должны таковыми и оставаться. CSS не должен быть написан внутри HTML-документа и наоборот. Как правило, HTML всегда будет представлять содержимое, а CSS всегда будет определять его оформление.

*Листинг 2. Пример кода HTML*

<!DOCTYPE html>

<html>

<head>

    <meta charset="utf-8">

    <title>Экспертная система</title>

    <link rel="stylesheet" href="es.css">

</head>

<body>

    <br><div id="title"></div>

    <br><div class="help" id="help"></div>

    <table id="dialog">

        <tr id="reload">

            <td class="quest" id="quest"></td>

        </tr>

        <tr class="line line0">

            <td class="answer" id="answer0"></td>

        </tr>

</table>

    <div>

    <br><img src="" id="img" onClick="window.location.reload()">

    <body  background="img/3.png">

    </div>

    <script src="db.js"></script> <!--база знаний-->

    <script src="es\_bin.js"></script> <!--машина вывода-->

</body>

</html>

*Листинг 3. Пример кода CSS*

body {

    background-color: rgb(15, 35, 28);

    color: rgb(223, 148, 8);

    font-family: "Comic Sans MS";

    font-size: 24px;

}

table {

    width: 45%;

    background-color: rgba(252, 250, 216, 0);

    border-width: 1,5px;

}

img {

    border-radius: 1px;

    width: 22%;

    height: auto;

    position: absolute;

        left: 150px;

        top: 260px;

 Для написания сценария для HTML-страницы использовался – JavaScript.

JavaScript ("JS" для краткости) — это полноценный динамический язык программирования, который применяется к HTML документу, и может обеспечить динамическую интерактивность на веб-сайтах.

JavaScript не предназначен для создания автономных приложений. Программа на JavaScript встраивается непосредственно в исходный текст HTML-документа и интерпретируется браузером по мере загрузки этого документа.

Изначально при открытии HTML страницы можно увидеть первый вопрос экспертной системы и ответы на него, вспомогательное предложение и название самой экспертной системы. Для дальнейшей работы приложения требуется выбрать вариант ответа, после чего появляется новый вопрос с вариантами ответов из файла JSON.

Код, который обрабатывает ответы (объекты) и выводит в последующем новый вопрос (ключ) с вариантами ответов находится в файле JavaScript .

Если пользователь дойдет до конечного вопроса, на который не будет вариантов ответов, будет выведено изображение с ответом и работа экспертной системы будет приостановлена. Что бы обнулить результат экспертной системы пользователю достаточно будет нажать на изображение (это описано в вспомогательном предложении).

*Листинг 3. Пример кода JavaScript*

let \_quest = document.querySelector('#quest'); *// элемент вопроса*

let \_title = document.querySelector('#title'); *// элемент заголовка*

let \_help = document.querySelector('#help'); *// элемент подсказки*

const init = function () { *// инициализация*

    run = true; *// запускаем приложение*

    \_title.innerHTML = es.title;

    \_help.innerHTML = 'Кликни на картинку для перехода в начало ЭС.';

    print\_dialog(es.start); *// печатаем первый вопрос и ответы*

};

const print\_dialog = function (post) { *// варианты ответов*

    document

        .querySelectorAll('.line') *// все по имени класса*

        .forEach(line => line.hidden = true); *// делаем скрытыми*

    let answers = es.dict[post]; *// узнаем текущие ответы*

    if (typeof answers === 'undefined') { *// если уже достигнут листок дерева*

        run = false; *// остановить работу ЭС*

        \_quest.innerHTML = `Выбор сделан -> ${post.toUpperCase()}`; *// публикуем выбор*

        document.getElementById('img').src = "img/" + post + ".png"; *//выводим изображение*

    }

# Заключение

В данном проекте мы рассмотрели создание экспертной системы при помощи языка разметки гипертекста HTML, каскадных таблиц CSS, JSON и языка программирования JavaScript.

Создание проекта с помощью такого варианта, позволяет эксперту изменять только само дерево решений, не меняя код самой программы, что очень удобно для эксперта, не знающего языки программирования.

Но в данном проекте так же есть недостатки, так пользователь может не проходить всю экспертную систему, а просто посмотреть весь код страницы, на которой вся информация на виду. Пользователь может узнать все ответы и вопросы из дерева решений. Данный минус еще предстоит изучить в будущем, и доработать программу.

# Список литературы

1. Билл Кеннеди, Чак Муссиано - "HTML и XHTML. Подробное руководство (HTML & HXTML. The Definitive Guide)"
2. Эрик Мейер - "CSS-каскадные таблицы стилей. Подробное руководство (Cascading Style Sheets: The Definitive Guide)"
3. К. Шмитт - "CSS. Рецепты программирования (CSS: Cookbook)"
4. Б. Хеник - "HTML и CSS. Путь к совершенству (HTML и CSS: The Good Parts)"
5. Дэвид Флэнаган - "JavaScript. Подробное руководство (JavaScript. The Definitive Guide)"

# Приложения

# *Приложение 1. Код HTML (index.html)*

<!DOCTYPE html>

<html>

<head>

    <meta charset="utf-8">

    <title>Экспертная система</title>

    <link rel="stylesheet" href="es.css">

</head>

<body>

    <br><div id="title"></div>

    <br><div class="help" id="help"></div>

    <table id="dialog">

        <tr id="reload">

            <td class="quest" id="quest"></td>

        </tr>

        <tr class="line line0">

            <td class="answer" id="answer0"></td>

        </tr>

        <tr class="line line1">

            <td class="answer" id="answer1"></td>

        </tr>

        <tr class="line line2">

            <td class="answer" id="answer2"></td>

        </tr>

        <tr class="line line3">

            <td class="answer" id="answer3"></td>

        </tr>

        <tr class="line line4">

            <td class="answer" id="answer4"></td>

        </tr>

        <tr class="line line5">

            <td class="answer" id="answer5"></td>

        </tr>

    </table>

    <div>

    <br><img src="" id="img" onClick="window.location.reload()">

    <body  background="img/3.png">

    </div>

    <script src="db.js"></script> <!--база знаний-->

    <script src="es\_bin.js"></script> <!--машина вывода-->

</body>

</html>

# *Приложение 2. Код CSS (es.c*ss*)*

body {

    background-color: rgb(15, 35, 28);

    color: rgb(223, 148, 8);

    font-family: "Comic Sans MS";

    font-size: 24px;

}

table {

    width: 45%;

    background-color: rgba(252, 250, 216, 0);

    border-width: 1,5px;

}

td {

    padding-left: 10px;

    height: 100px;

    border-color: rgb(11, 185, 98);

    border-width: 1px;

    border-style: solid;

    text-align: center;

}

td.quest {

    height: 90px;

    background-color: rgba(241, 238, 185, 0.219);

    color: rgb(255, 145, 1)

}

img {

    border-radius: 1px;

    width: 22%;

    height: auto;

    position: absolute;

        left: 150px;

        top: 260px;

}

td.answer {

    height: 60px;

}

div {

    text-align: left;

}

div.help {

    color: rgb(125, 179, 55);

    font-size: 16px;

    margin-bottom: 20px;

    padding-left: 150px;

}

# *Приложение 3. Код JSON (db.js)*

let es = {

    "title": "ЭС может угадать загаданного героя из сериала Ривердейл",

    "start": "Роль героя",

    "dict": {

        "Роль героя": ["Главная", "Второстепенная"],

        "Главная": ["школьник", "родитель"],

        "школьник": ["мужской пол", "женский пол"],

        "мужской пол": ["блондин", "рыжий", "брюнет"],

        "блондин": [ "такого героя нет"],

        "брюнет": [ "Джагхед Джонс"],

        "рыжий": [ "Арчибальд Эндрюс"],

        "женский пол": ["блондинка", "рыжая", "брюнетка"],

        "блондинка": [ "Элизабет Купер"],

        "брюнетка": [ "Вероника Лодж"],

        "рыжая": [ "Шерил Блоссом"],

        "родитель": ["отрицательная роль", "положительная роль","переход от отрицательной к положительной роли"],

        "отрицательная роль": [ "Хайрам Лодж"],

        "положительная роль": ["Люк Перри" ],

        "переход от отрицательной к положительной роли": [ "ЭфПи Джонс"],

        "Второстепенная": ["Родитель", "Школьник", "бабушка"],

        "Родитель": ["Положительный герой", "Отрицательный герой"],

        "Положительный герой": [ "Мери Эндрюс"],

        "Отрицательный герой": [ "Хэл Купер"],

        "Школьник": ["положительный герой", "отрицательный герой", "нейтральный герой"],

        "положительный герой": ["Джозефина Маккой" ],

        "отрицательный герой": ["Сара Дежарден"],

        "нейтральный герой": ["Тина Патель"],

        "бабушка": ["Роуз Блоссом" ]

    }

}

# *Приложение 4. Код JavaScript (es\_bin.js)*

// global variables

let colors = ['rgba(252, 250, 216, 0)', '#464']; // массив цветов выкл/вкл

let run; // запущено или остановлено приложение

let \_quest = document.querySelector('#quest'); // элемент вопроса

let \_title = document.querySelector('#title'); // элемент заголовка

let \_help = document.querySelector('#help'); // элемент подсказки

// functions

const init = function () { // инициализация

    run = true; // запускаем приложение

    \_title.innerHTML = es.title;

    \_help.innerHTML = 'Кликни на картинку для перехода в начало ЭС.';

    print\_dialog(es.start); // печатаем первый вопрос и ответы

};

const print\_dialog = function (post) { // варианты ответов

    document

        .querySelectorAll('.line') // все по имени класса

        .forEach(line => line.hidden = true); // делаем скрытыми

    let answers = es.dict[post]; // узнаем текущие ответы

    if (typeof answers === 'undefined') { // если уже достигнут листок дерева

        run = false; // остановить работу ЭС

        \_quest.innerHTML = `Выбор сделан -> ${post.toUpperCase()}`; // публикуем выбор

        document.getElementById('img').src = "img/" + post + ".png";

    }

    else {

        if (answers.length === 0) { // если нет ответов на вопрос

            run = false; // остановить работу ЭС

            \_quest.innerHTML = `Для категории "${post}" нет выбора.`;

        }

        else {  // публикуем вопрос и варианты ответов

            \_quest.innerHTML = `Сделайте выбор из категории<br>"${post}":`;

            answers

                .forEach((answer, index) => {

                    document.querySelector('#answer' + String(index)).innerHTML = answer;

                    document.querySelector('.line' + String(index)).hidden = false;

                });

        }

    }

}

document // при загрузке страницы

    .addEventListener("DOMContentLoaded", init);

document // при клике по ячейке таблицы

    .querySelectorAll('#dialog .answer') // найти массив ячеек таблицы с ответами

    .forEach(td\_answer => { // для каждой ячейки назначить обработчики событий

        td\_answer.addEventListener("click", () => print\_dialog(td\_answer.innerHTML));

        td\_answer.addEventListener('mouseenter', () => td\_answer.style.backgroundColor = colors[1]);

        td\_answer.addEventListener('mouseleave', () => td\_answer.style.backgroundColor = colors[0]);

    });

РЕЦЕНЗИЯ

на курсовой проект \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

доцент кафедры ИТиПИ, к.т.н., доцент

Беляков Андрей Юрьевич