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 Программная информатика

шифр ПИб-1328-2019

Мясников Павел Александрович

Проверил: доцент кафедры ИТиПИ,

к.т.н., доцент Беляков Андрей Юрьевич

Оглавление

[Постановка задачи 2](#_Toc93853011)

[Разработка структуры для хранения данных 4](#_Toc93853012)

[Программная реализация 6](#_Toc93853013)

[Заключение 10](#_Toc93853014)

[Список литературы 11](#_Toc93853015)

[Приложение 12](#_Toc93853016)

постановка задачи

Имеется следующая ситуация: молодой человек в первый раз решает познакомиться с игрой, а именно узнать, какие герои есть в этой игре.

Для решения поставленной задачи нужно разработать экспертную систему, которая будет показывать, какие герои, наиболее подходящие для новичков, присутствуют в игре.

Экспертная система разрабатывалась в системе программирования Visual Studio Code, на таком языке программирования, как JavaScript, так как этот язык отлично подходит для разработки экспертной системы с дальнейшим применением в практической деятельности.

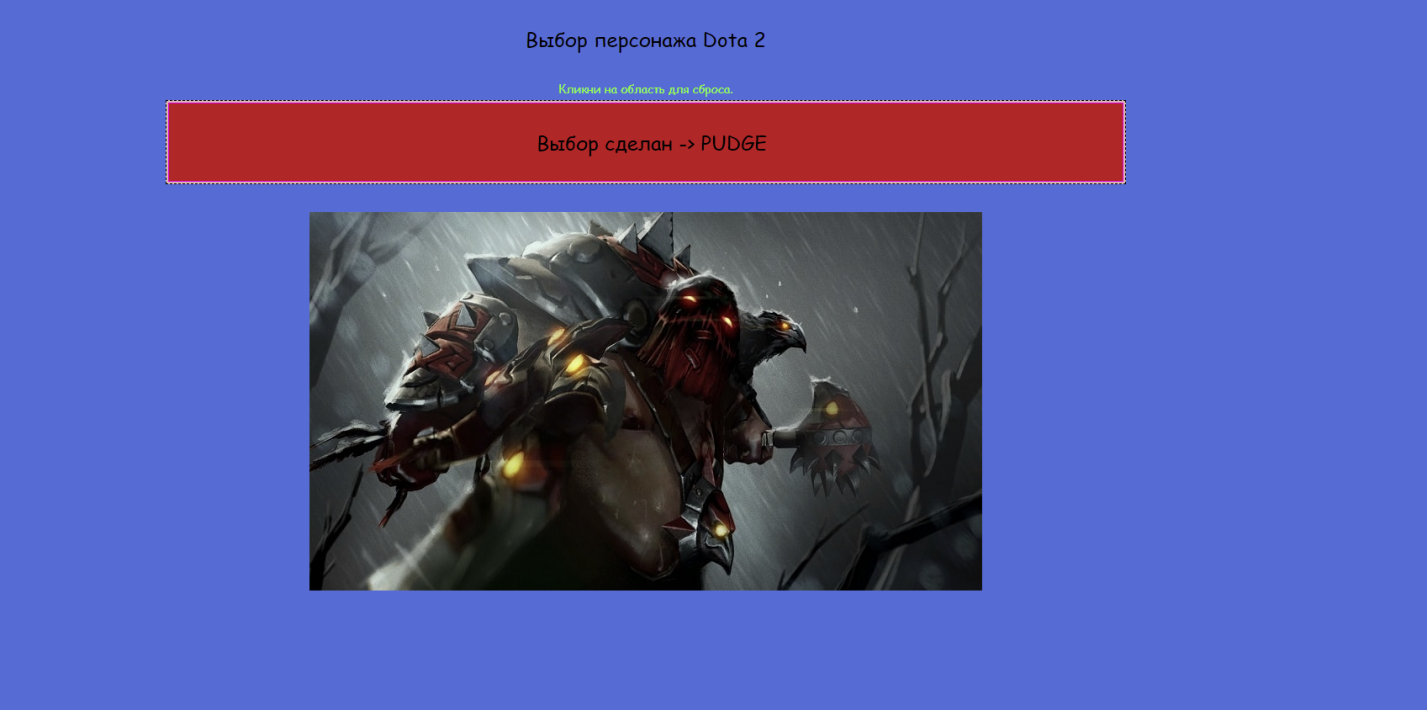
Также для создания интерфейса пользователя использовались языки html и css.

На изображении 1 можно увидеть, как выглядит стартовая форма, разработанная для поставленной задачи в системе программирования Visual Studio Code.



Изображение 1. Стартовая форма

На изображении 2 можно увидеть, как выглядит результат выполнения данной программы в рамках поставленной задачи.



Изображение 2. Результат

Разработка структуры для хранения данных

В базе знаний структура хранений данных разработана в формате JSON. JSON – это текстовый формат обмена данными, основанный на JavaScript. Несмотря на происхождение от JavaScript, формат считается независимым от языка и может использоваться практически с любым [языком программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F" \o "Язык программирования). Для многих языков существует готовый код для создания и обработки данных в формате JSON.

Объект JSON это формат данных — ключ-значение, который обычно рендерится в фигурных скобках. Пример показан на листинге 1 (полный листинг находится в Приложении 1).

"title": "Выбор персонажа Dota 2",

    "start": "Какую позицию вы выберите ",

    "dict": {

        "Какую позицию вы выберите ": ["Mid line","Safe line","Hard line","Soft support","Hard support"],

        "Mid line":["Выберите легкого персонажа","Выберите среднего пресонажа ","Выберите тяжелого персонажа"],

        "Safe line":["Выберите Легкого персонажа","Выберите Среднего пресонажа ","Выберите Тяжелого персонажа"],

        "Hard line":["Выберите Легкого Персонажа","Выберите Среднего Пресонажа ","Выберите Тяжелого Персонажа"],

        "Soft support":["Выбрать легкого персонажа","Выбрать среднего пресонажа ","Выбрать тяжелого персонажа"],

        "Hard support":["Выбрать Легкого персонажа","Выбрать Среднего пресонажа ","Выбрать Тяжелого персонажа"],

        "Выберите легкого персонажа":["Tinker","Pudge","Sniper"],

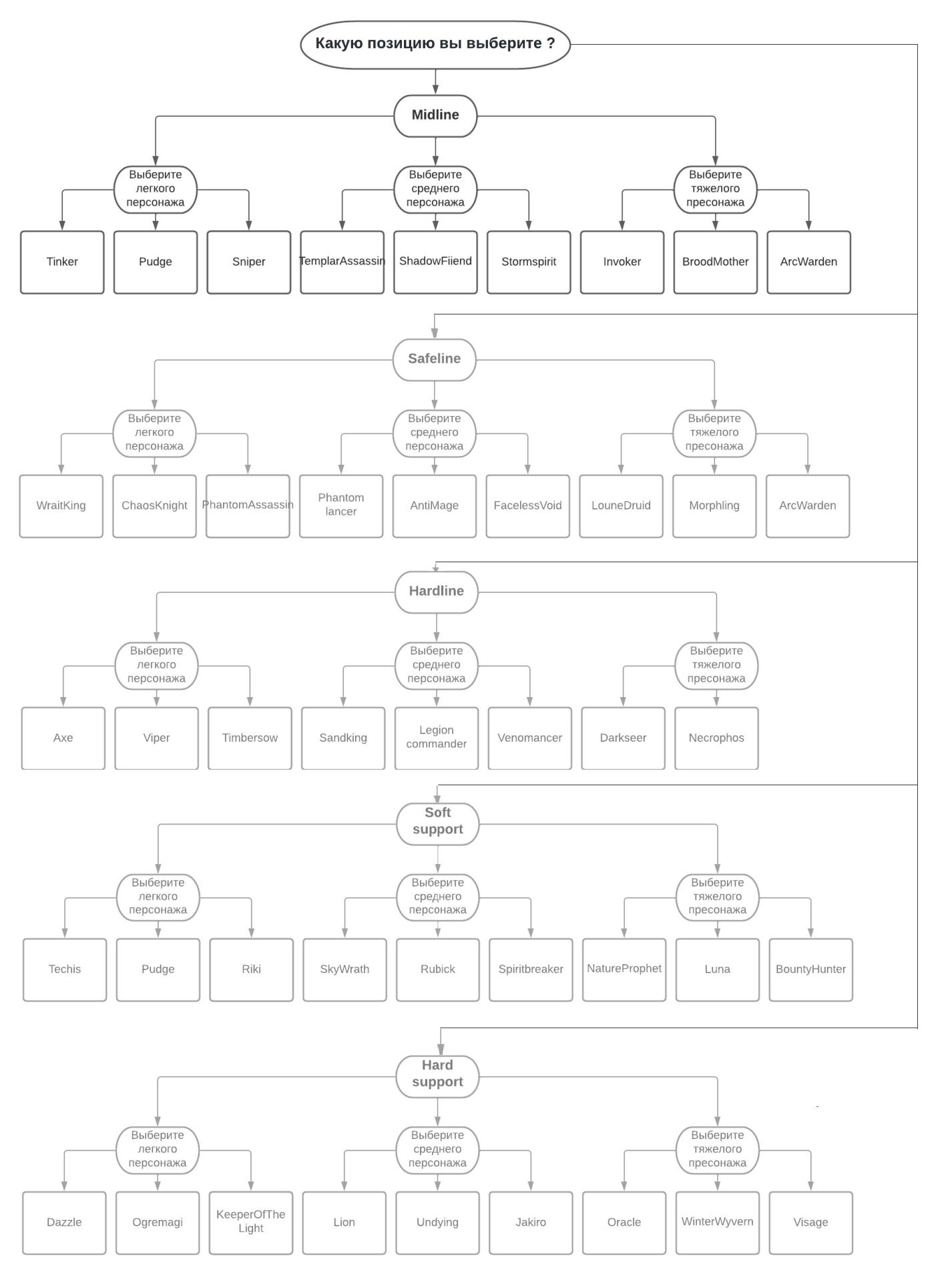
        "Выберите среднего пресонажа ":["TemplarAssassin","ShadowFind","StormSpirit"],

        "Выберите тяжелого персонажа":["Invoker","BroodMother","ArcWarden"],

Листинг 1 – Пример кода JSON

Пары ключ-значение разделены двоеточием, например "key" : "value". Каждая пара значений разделена двоеточием, таким образом, середина JSON выглядит так: "key" : "value", "key" : "value", "key" : "value".

На изображении 3 представлено дерево решений экспертной системы.



Изображение 3. Дерево решений

программная реализация

При создании продукционной экспертной системы была создана машина вывода, которая занимается обработкой событий выбора из значений ответа на текущий вопрос.

Были объявлены глобальные переменные, которые необходимы для взаимодействия с базой знаний и интерфейсом пользователя, они представлены на листинге 2 (полный листинг находится в Приложении 2).

let colors = ['', '#777']; // 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

Листинг 2 – Глобальные переменные, объявленные в модуле es\_bin.js

На листинге 3 можно увидеть код функции запуска приложения, а также выведение на экран первого вопроса и заголовка страницы (полный листинг находится в Приложении 2).

const init = function () { // инициализация

    run = true; // запускаем приложение

    \_title.innerHTML = es.title;

    \_help.innerHTML = 'Кликни на область для сброса.';

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

};

Листинг 3 – код запуска приложения модуля es\_bin.js

На листинге 4 представлен код реализации обработчика событий (полный листинг находится в Приложении 2). Обрабатываются все варианты ответов и задаётся переменная с текущими ответами.

В случае, если вариантов ответов больше нет, выдаётся вывод, а также меняется картинка в соответствии с ответом. Далее, если в данной категории не будет конечного ответа, то тогда идёт вывод на интерфейсе пользователя сообщение о том, что в данной категории ответов нет. Дальше представлен код, отвечающий за выведение на интерфейс пользователя вопрос и варианты ответов.

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 + ".jpg";

    }

    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;

                });

        }

    }

}

Листинг 4 – часть программного кода модуля es\_bin.js

Интерфейс приложения разработан с помощью языка гипертекстовой разметки HTML и каскадных таблиц стилей CSS. HTML позволяет создавать и структурировать разделы, параграфы, заголовки, ссылки и блоки для веб-страниц и приложений.

HTML не является языком программирования, то есть он не имеет возможности создавать динамические функции.

При работе с HTML используются простые структуры кода (теги и атрибуты), чтобы разметить страницу веб-сайта.

CSS служит для описания оформления внешнего вида документа, созданного с использованием языка разметки (HTML, XHTML, XML).

Назначение CSS – отделять то, что задает внешний вид страницы, от ее содержания. Если документ создан только с использованием HTML, то в нем определяется не только каждый элемент, но и способ его отображения (цвет, шрифт, положение блока и т. д.). Если же подключены каскадные таблицы стилей, то HTML описывает только очередность объектов. А за все их свойства отвечает CSS. В HTML достаточно прописывать класс, не перечисляя все стили каждый раз. Пример использования HTML представлен на листинге 5 (полный листинг находится в Приложении 4).

<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>

Листинг 5 – пример программного кода модуля index.htnl

Пример использования CCS представлен на листинге 6 (Полный листинг находится в Приложении 3).

body {

    background-color: rgb(51, 118, 218); /\* Выбор цвета основного фона \*/

    color: rgb(0, 0, 0);  /\* Выбор цвета текста \*/

    font-family: "Comic Sans MS";

    font-size: 24px;

}

table {

    width: 60%;

    background-color: rgb(236, 182, 182); /\* Выбор цвета внутри таблицы  \*/

    border-width: 1px;

    border-style: dashed;

    border-color: rgb(0, 0, 0); /\* Выбор цвета обводки таблицы  \*/

    margin: auto;

}

td {

    padding-left: 15px;

    height: 100px;

    border-color: rgb(255, 0, 221); /\* Выбор цвета выдерления внутри таблицы \*/

    border-width: 1px;

    border-style: solid;

text-align: center;

Листинг 6 – пример программного кода модуля es.css

заключение

При выполнении данного проекта мной был проанализирован материал о персонажах, присутствующих в игре Dota 2.

На основании полученной о персонажах информации была возведена база знаний экспертной системы, для создания которой были изучены технологии построения базы знаний экспертной системы.

В процессе написания курсовой работы были изучены механизмы логического вывода, а также повторен материал создания экспертной системы в системе программирования Visual Studio Code.

Продуктом курсового проекта стала разработанная мной модель базы знаний экспертной системы «Выбор персонажа Dota 2». Экспертная система была изобретена в виде веб-приложения на языках JavaScript, HTML, CSS.

Как итог, могу сказать, что экспертная система наглядно показала свою работоспособность и результативность в рамках поставленной задачи.

Однако у данной программы имеются недостатки, к которым можно отнести малый объём базы знаний и несложный дизайн. Соответственно, для реализации разработанного проекта в практической деятельности необходимо расширить базу знаний и разработать более красивый дизайн.

Список литературы

1. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский - СПб: Питер, 2000 - 384 с.;
2. Гаскаров, Д.Б. Интеллектуальные информационные системы. - М.: Высшая школа, 2003.;
3. Интернет ресурс – Браузер: документ, события, интерфейсы URL: https://learn.javascript.ru/ui (дата обращения 3.04.2022);
4. Интернет ресурс – Справочник CSS URL: http://htmlbook.ru/css/help (дата обращения 3.04.2022);
5. Интернет ресурс – Браузер: https://dota2.fandom.com/wiki/Dota\_2\_Wiki (дата обращения 3.04.2022).

Приложение

Приложение 1 – Структура хранения данных JSON

let es = {

    "title": "Выбор персонажа Dota 2",

    "start": "Какую позицию вы выберите ",

    "dict": {

        "Какую позицию вы выберите ": ["Mid line","Safe line","Hard line","Soft support","Hard support"],

        "Mid line":["Выберите легкого персонажа","Выберите среднего пресонажа ","Выберите тяжелого персонажа"],

        "Safe line":["Выберите Легкого персонажа","Выберите Среднего пресонажа ","Выберите Тяжелого персонажа"],

        "Hard line":["Выберите Легкого Персонажа","Выберите Среднего Пресонажа ","Выберите Тяжелого Персонажа"],

        "Soft support":["Выбрать легкого персонажа","Выбрать среднего пресонажа ","Выбрать тяжелого персонажа"],

        "Hard support":["Выбрать Легкого персонажа","Выбрать Среднего пресонажа ","Выбрать Тяжелого персонажа"],

        "Выберите легкого персонажа":["Tinker","Pudge","Sniper"],

        "Выберите среднего пресонажа ":["TemplarAssassin","ShadowFind","StormSpirit"],

        "Выберите тяжелого персонажа":["Invoker","BroodMother","ArcWarden"],

        "Выберите Легкого персонажа":["WraitKing","ChaosKnight","PhantomAssassin"],

        "Выберите Среднего пресонажа ":["PhantomLancer","Antimage","FacelessVoid"],

        "Выберите Тяжелого персонажа":["LounDruide","ArcWarden","Morphling"],

        "Выберите Легкого Персонажа":["Axe","Viper","Timbersow"],

        "Выберите Среднего Пресонажа ":["SandKing","Legioncommander","Venomancer"],

        "Выберите Тяжелого Персонажа":["Darkseer","Necrophos"],

        "Выбрать легкого персонажа":["Techis","Pudge","Riki"],

        "Выбрать среднего пресонажа ":["SkyWrath","Rubick","SpiritBreaker"],

        "Выбрать тяжелого персонажа":["NatureProphet","Luna","BountyHunter"],

        "Выбрать Легкого персонажа":["Dazzle","Ogremagi","KeeperOfTheLight"],

        "Выбрать Среднего пресонажа ":["Lion","Undying","Jakiro"],

        "Выбрать Тяжелого персонажа":["Oracle","WinterWyvern","Visage"],

    }

}

Приложение 2 – Текст программы на языке JavaScript

let colors = ['', '#777']; // 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 область для сброса.';

    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 + ".jpg";

    }

    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;

                });

        }

    }

}

// event handlers

document // при загрузке страницы

    .addEventListener("DOMContentLoaded", init);

document // при клике по заголовку таблицы - начинаем сначала

    .querySelector('#reload')

    .addEventListener('click', 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]);

    });

Приложение 3 – Оформление страницы CSS

body {

    background-color: rgb(51, 118, 218); /\* Выбор цвета основного фона \*/

    color: rgb(0, 0, 0);  /\* Выбор цвета текста \*/

    font-family: "Comic Sans MS";

    font-size: 24px;

}

table {

    width: 60%;

    background-color: rgb(236, 182, 182); /\* Выбор цвета внутри таблицы  \*/

    border-width: 1px;

    border-style: dashed;

    border-color: rgb(0, 0, 0); /\* Выбор цвета обводки таблицы  \*/

    margin: auto;

}

td {

    padding-left: 15px;

    height: 100px;

    border-color: rgb(255, 0, 221); /\* Выбор цвета выдерления внутри таблицы \*/

    border-width: 1px;

    border-style: solid;

    text-align: center;

}

td.quest {

    height: 90px;

    background-color: rgb(206, 24, 24); /\*Выбор цвета заголовка таблицы\*/

    color: rgb(114, 0, 0)5);  /\*Выбор цвета выдерления заголовка  таблицы\*/

}

td.answer {

    height: 60px;

}

div {

    text-align: center;

}

div.help {

    color: rgb(60, 255, 0);  /\*Выбор цвета текста помощи \*/

    font-size: 14px;

    margin-bottom: 4px;

}

Приложение 4 – Разметка страницы 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="\img\1.jpg" id ="img" onclick="window.location.reload()"/>

    </div>

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

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

</body>

</html>

РЕЦЕНЗИЯ

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

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

доцент кафедры ИТиПИ, к.т.н., доцент Беляков Андрей Юрьевич