**Электронная обучающая система по технике безопасности**

**«Онлайн обучение»**

**РУКОВОДСТВО ПРОГРАММИСТА**

Листов 20

2021

СОДЕРЖАНИЕ

[1 НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ ПРОГРАММЫ 3](#_Toc71798934)

[1.1 Назначение программы 3](#_Toc71798935)

[1.2 Функции, выполняемые программой 3](#_Toc71798936)

[1.3 Условия, необходимые для выполнения программы 3](#_Toc71798937)

[1.3.1 Объем оперативной памяти 3](#_Toc71798938)

[1.3.2 Требования к составу периферийных устройств 3](#_Toc71798939)

[1.3.3 Требования к параметрам периферийных устройств 3](#_Toc71798940)

[1.3.4 Требования к программному обеспечению 4](#_Toc71798941)

[1.3.5 Требования к персоналу (программисту) 4](#_Toc71798942)

[2 ХАРАКТЕРИСТИКА ПРОГРАММЫ 4](#_Toc71798943)

[2.1 Описание основных характеристик программы 4](#_Toc71798944)

[2.1.1 Режим работы программы 4](#_Toc71798945)

[2.2 Описание основных особенностей программы 4](#_Toc71798946)

[2.2.1 Самовосстанавливаемость программы 4](#_Toc71798947)

[3 ОБРАЩЕНИЕ К ПРОГРАММЕ 5](#_Toc71798948)

[3.1 Руководство программиста 5](#_Toc71798949)

[4 ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ](#_Toc71798955) 20

[4.1 Организация используемой входной и выходной информации](#_Toc71798956) 20

# 1 НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ ПРОГРАММЫ

# 1.1 Назначение программы

Электронная обучающая система предназначен для обучения сотрудников предприятия и проверки их знаний.

# 1.2 Функции, выполняемые программой

Приложение обладает следующими возможностями:

- Обучение.

- Проверка знаний.

# 1.3 Условия, необходимые для выполнения программы

# 1.3.1 Объем оперативной памяти

Рекомендуемый объем оперативной памяти 200 Мб или выше.

# 1.3.2 Требования к составу периферийных устройств

Для полноценного взаимодействия с программой необходима следующая периферия:

* мышка;
* клавиатура;
* монитор.
* Интернет-подключение

# 1.3.3 Требования к параметрам периферийных устройств

Подключаемые периферийные устройства должны быть IBM-совместимыми.

# 1.3.4 Требования к программному обеспечению

Для работы программы на компьютере должна быть установлена windows 7 или выше.

# 1.3.5 Требования к персоналу (программисту)

Программист должен иметь минимум среднее техническое образование и должен быть аттестован минимум на II квалификационную группу по электробезопасности.

В перечень задач, выполняемых программистом, должны входить:

* задача поддержания работоспособности технических средств;
* задача поддержания работоспособности системных программных средств операционной системы.

# 2 ХАРАКТЕРИСТИКА ПРОГРАММЫ

# 2.1 Описание основных характеристик программы

# 2.1.1 Режим работы программы

Электронная обучающая система работает в стандартном режиме – оконный режим работы в операционной системе.

# 2.2 Описание основных особенностей программы

# 2.2.1 Самовосстанавливаемость программы

Самовосстанавливаемость специальной электронной обучающей системы обеспечивается стандартными средствами операционной системы.

# 3 ОБРАЩЕНИЕ К ПРОГРАММЕ

Основной файл Script.js отвечает за все функции в системе.

Его код представлен ниже:

Константы вписаные с запросом к параметрам.

const start\_btn = document.querySelector(".start\_btn button");

const info\_box = document.querySelector(".info\_box");

const exit\_btn = info\_box.querySelector(".buttons .quit");

const continue\_btn = info\_box.querySelector(".buttons .restart");

const quiz\_box = document.querySelector(".quiz\_box");

const result\_box = document.querySelector(".result\_box");

const option\_list = document.querySelector(".option\_list");

const time\_line = document.querySelector("header .time\_line");

const timeText = document.querySelector(".timer .time\_left\_txt");

const timeCount = document.querySelector(".timer .timer\_sec");

Вызов окна с правилами при нажатии кноки «Начать тест»

start\_btn.onclick = ()=>{

info\_box.classList.add("activeInfo");

}

Закрытие окна с правилами при нажатии кнопки «Покинуть тест»

exit\_btn.onclick = ()=>{

info\_box.classList.remove("activeInfo");

}

Открытия окна с тестом и запуск таймера при нажатии кнопки «Продолжить»

continue\_btn.onclick = ()=>{

info\_box.classList.remove("activeInfo");

quiz\_box.classList.add("activeQuiz");

showQuetions(0);

queCounter(1);

startTimer(15);

startTimerLine(0);

}

let timeValue = 15;

let que\_count = 0;

let que\_numb = 1;

let userScore = 0;

let counter;

let counterLine;

let widthValue = 0;

Константы вписаные с запросом к параметрам.

const restart\_quiz = result\_box.querySelector(".buttons .restart");

const quit\_quiz = result\_box.querySelector(".buttons .quit");

Открытие теста при нажатии кнопки «Повторить».

restart\_quiz.onclick = ()=>{

quiz\_box.classList.add("activeQuiz"); //show quiz box

result\_box.classList.remove("activeResult"); //hide result box

timeValue = 15;

que\_count = 0;

que\_numb = 1;

userScore = 0;

widthValue = 0;

showQuetions(que\_count); Вызов функции showquestion

queCounter(que\_numb); //passing que\_numb value to queCounter

clearInterval(counter); //clear counter

clearInterval(counterLine); //clear counterLine

startTimer(timeValue); //calling startTimer function

startTimerLine(widthValue); //calling startTimerLine function

timeText.textContent = "Осталось времени"; //change the text of timeText to Time Left

next\_btn.classList.remove("show"); //hide the next button

}

Закрытие теста при нажатии кнопки «Покинуть тест»

// if quitQuiz button clicked

quit\_quiz.onclick = ()=>{

window.location.reload(); //reload the current window

}

const next\_btn = document.querySelector("footer .next\_btn");

const bottom\_ques\_counter = document.querySelector("footer .total\_que");

Вывод следующего вопроса при нажатии кнопки «Далее»

next\_btn.onclick = ()=>{

if(que\_count < questions.length - 1){ //if question count is less than total question length

que\_count++; //increment the que\_count value

que\_numb++; //increment the que\_numb value

showQuetions(que\_count); //calling showQestions function

queCounter(que\_numb); //passing que\_numb value to queCounter

clearInterval(counter); //clear counter

clearInterval(counterLine); //clear counterLine

startTimer(timeValue); //calling startTimer function

startTimerLine(widthValue); //calling startTimerLine function

timeText.textContent = "Осталось времени"; //change the timeText to Time Left

next\_btn.classList.remove("show"); //hide the next button

}else{

clearInterval(counter); //clear counter

clearInterval(counterLine); //clear counterLine

showResult(); //calling showResult function

}

}

Получение вопросов и ответов из массива

function showQuetions(index){

const que\_text = document.querySelector(".que\_text");

//creating a new span and div tag for question and option and passing the value using array index

let que\_tag = '<span>'+ questions[index].numb + ". " + questions[index].question +'</span>';

let option\_tag = '<div class="option"><span>'+ questions[index].options[0] +'</span></div>'

+ '<div class="option"><span>'+ questions[index].options[1] +'</span></div>'

+ '<div class="option"><span>'+ questions[index].options[2] +'</span></div>'

+ '<div class="option"><span>'+ questions[index].options[3] +'</span></div>';

que\_text.innerHTML = que\_tag; //adding new span tag inside que\_tag

option\_list.innerHTML = option\_tag; //adding new div tag inside option\_tag

const option = option\_list.querySelectorAll(".option");

// set onclick attribute to all available options

for(i=0; i < option.length; i++){

option[i].setAttribute("onclick", "optionSelected(this)");

}

}

// creating the new div tags which for icons

let tickIconTag = '<div class="icon tick"><i class="fas fa-check"></i></div>';

let crossIconTag = '<div class="icon cross"><i class="fas fa-times"></i></div>';

Проверка ответа если пользователь выбрал ответ.

function optionSelected(answer){

clearInterval(counter); //clear counter

clearInterval(counterLine); //clear counterLine

let userAns = answer.textContent; //getting user selected option

let correcAns = questions[que\_count].answer; //getting correct answer from array

const allOptions = option\_list.children.length; //getting all option items

if(userAns == correcAns){ //if user selected option is equal to array's correct answer

userScore += 1; //upgrading score value with 1

answer.classList.add("correct"); //adding green color to correct selected option

answer.insertAdjacentHTML("beforeend", tickIconTag); //adding tick icon to correct selected option

console.log("Правильно");

console.log("Правильных ответов = " + userScore);

}else{

answer.classList.add("incorrect"); //adding red color to correct selected option

answer.insertAdjacentHTML("beforeend", crossIconTag); //adding cross icon to correct selected option

console.log("Неправильно");

for(i=0; i < allOptions; i++){

if(option\_list.children[i].textContent == correcAns){ //if there is an option which is matched to an array answer

option\_list.children[i].setAttribute("class", "option correct"); //adding green color to matched option

option\_list.children[i].insertAdjacentHTML("beforeend", tickIconTag); //adding tick icon to matched option

console.log("Автоматически правильный ответ.");

}

}

}

for(i=0; i < allOptions; i++){

option\_list.children[i].classList.add("disabled"); //once user select an option then disabled all options

}

next\_btn.classList.add("show"); //show the next button if user selected any option

}

function showResult(){

info\_box.classList.remove("activeInfo"); //hide info box

quiz\_box.classList.remove("activeQuiz"); //hide quiz box

result\_box.classList.add("activeResult"); //show result box

const scoreText = result\_box.querySelector(".score\_text");

if (userScore > 3){ // if user scored more than 3

//creating a new span tag and passing the user score number and total question number

let scoreTag = '<span>Поздравляем! 🎉, У вас <p>'+ userScore +'</p> из <p>'+ questions.length +'</p></span>';

scoreText.innerHTML = scoreTag; //adding new span tag inside score\_Text

}

else if(userScore > 1){ // if user scored more than 1

let scoreTag = '<span>отлично 😎, У вас <p>'+ userScore +'</p> из<p>'+ questions.length +'</p></span>';

scoreText.innerHTML = scoreTag;

}

else{ // if user scored less than 1

let scoreTag = '<span>плохо 😐, у вас <p>'+ userScore +'</p> из <p>'+ questions.length +'</p></span>';

scoreText.innerHTML = scoreTag;

}

}

function startTimer(time){

counter = setInterval(timer, 1000);

function timer(){

timeCount.textContent = time; //changing the value of timeCount with time value

time--; //decrement the time value

if(time < 9){ //if timer is less than 9

let addZero = timeCount.textContent;

timeCount.textContent = "0" + addZero; //add a 0 before time value

}

if(time < 0){ //if timer is less than 0

clearInterval(counter); //clear counter

timeText.textContent = "Время вышло"; //change the time text to time off

const allOptions = option\_list.children.length; //getting all option items

let correcAns = questions[que\_count].answer; //getting correct answer from array

for(i=0; i < allOptions; i++){

if(option\_list.children[i].textContent == correcAns){ //if there is an option which is matched to an array answer

option\_list.children[i].setAttribute("class", "option correct"); //adding green color to matched option

option\_list.children[i].insertAdjacentHTML("beforeend", tickIconTag); //adding tick icon to matched option

console.log("Время вышло: выбран автоматически правильный ответ.");

}

}

for(i=0; i < allOptions; i++){

option\_list.children[i].classList.add("disabled"); //once user select an option then disabled all options

}

next\_btn.classList.add("show"); //show the next button if user selected any option

}

}

}

function startTimerLine(time){

counterLine = setInterval(timer, 29);

function timer(){

time += 1; //upgrading time value with 1

time\_line.style.width = time + "px"; //increasing width of time\_line with px by time value

if(time > 549){ //if time value is greater than 549

clearInterval(counterLine); //clear counterLine

}

}

}

function queCounter(index){

//creating a new span tag and passing the question number and total question

let totalQueCounTag = '<span><p>'+ index +'</p> из <p>'+ questions.length +'</p> вопросов</span>';

bottom\_ques\_counter.innerHTML = totalQueCounTag; //adding new span tag inside bottom\_ques\_counter

}

Модуль questions.js

Создание массива

let questions = [

{

numb: 1,

question: " Какое определение понятия «охрана труда» будет верным?",

answer: "охрана труда — система сохранения жизни и здоровья работников в процессе трудовой деятельности, включающая в себя правовые, социально-экономические, организационно-технические, санитарно-гигиенические, лечебно-профилактические, реабилитационные и иные мероприятия;",

options: [

"охрана труда — система сохранения жизни и здоровья работников в процессе трудовой деятельности, включающая в себя правовые, социально-экономические, организационно-технические, санитарно-гигиенические, лечебно-профилактические, реабилитационные и иные мероприятия;",

"охрана труда — совокупность факторов производственной среды и трудового процесса, оказывающих влияние на работоспособность и здоровье людей;",

"охрана труда — это техника безопасности и гигиена труда.",

"охрана труда"

]

},

{

numb: 2,

question: "Вводный инструктаж по охране труда, первичный инструктаж на рабочем месте, повторный, внеплановый, целевой инструктажи.",

answer: "Cascading Style Sheet",

options: [

"Вводный инструктаж по охране труда, первичный инструктаж на рабочем месте, повторный, внеплановый, целевой инструктажи.",

"Вводный инструктаж по охране труда, первичный, повторный и внеплановый инструктажи на рабочем месте.",

"Первичный инструктаж на рабочем месте, повторный, внеплановый, целевой инструктажи.",

"Никакие"

]

},

{

numb: 3,

question: "О чем работник обязан немедленно известить своего руководителя?",

answer: "о всем перечисленном.",

options: [

"о любой ситуации, угрожающей жизни и здоровью людей;",

"о каждом несчастном случае, происшедшем в организации;",

"об ухудшении состояния своего здоровья;",

"о всем перечисленном."

]

},

{

numb: 4,

question: "Каким локальным нормативным актом устанавливается режим рабочего времени в организации?",

answer: "Правилами внутреннего трудового распорядка организации;",

options: [

"Правилами внутреннего трудового распорядка организации;",

"Распоряжением руководителя подразделения.",

"Никаким",

" "

]

},

{

numb: 5,

question: "Кто подлежит обучению по охране труда и проверке знания требований охраны труда?",

answer: "все работники организации, в т.ч. руководитель;",

options: [

"все работники организации, в т.ч. руководитель;",

"только работники, занятые на работах повышенной опасности;",

"только работники службы охраны труда и руководители подразделений.",

"только работники службы охраны"

]

},

{

numb: 6,

question: "Как оказать первую помощь при артериальном кровотечении у пострадавшего?",

answer: "Наложить жгут выше места повреждения.",

options: [

"Наложить давящую повязку.",

"Наложить жгут выше места повреждения.",

"Наложить согревающий компресс, обеспечить покой.",

"Продезенфицировать"

]

},

{

numb: 7,

question: "Кто обеспечивает разработку и утверждение инструкций по охране труда для работников организации?",

answer: "работодатель с учетом изложенного в письменном виде мнения выборного профсоюзного или иного уполномоченного работниками органа;",

options: [

"работодатель с учетом изложенного в письменном виде мнения выборного профсоюзного или иного уполномоченного работниками органа;",

"руководитель работ;",

"служба охраны труда.",

"уборщики"

]

},

{

numb: 8,

question: "Что необходимо сделать в первую очередь при поражении человека электрическим током? ",

answer: "Освободить пострадавшего от действия электрического тока.",

options: [

"Освободить пострадавшего от действия электрического тока.",

"Приступить к реанимации пострадавшего.",

"Оттащить пострадавшего за одежду не менее чем на 8 метров от места касания. проводом земли или от оборудования, находящегося под напряжением.",

"Позвонить в скорую помощь."

]

},

{

numb: 9,

question: "Каждый работник имеет право на (ТК РФ Статья 219):",

answer: "все ответы верны",

options: [

"рабочее место, соответствующее требованиям охраны труда;",

"обязательное социальное страхование от несчастных случаев на производстве;",

"отказ от выполнения работы в случае возникновения опасности для его жизни;",

"обеспечение средствами индивидуальной и коллективной защиты за счет средств работодателя;",

"все ответы верны",

]

},

{

numb: 10,

question: "непосредственный руководитель работ, прошедший в установленном порядке обучение и проверку знаний по охране труда, проводит инструктаж работникам до начала их самостоятельной работы;",

answer: "лицо, назначенное распоряжением работодателя, проводит инструктаж в течение месяца после приема работника в организацию.",

options: [

"непосредственный руководитель работ, прошедший в установленном порядке обучение и проверку знаний по охране труда, проводит инструктаж работникам до начала их самостоятельной работы;",

"специалист по охране труда проводит инструктаж до начала производственной деятельности работника;",

"лицо, назначенное распоряжением работодателя, проводит инструктаж в течение месяца после приема работника в организацию.",

"Никто"

]

},

{

numb: 11,

question: "роки проведения специального обучения по охране труда руководителей и специалистов организаций:",

answer: "не реже одного раза в 3 года.",

options: [

"не реже одного раза в 5 лет;",

"по мере необходимости;",

"не реже одного раза в 3 года.",

"раз в 7 лет"

]

},

{

numb: 12,

question: "В какие сроки проводится повторный инструктаж на рабочем месте?",

answer: "не реже одного раза в шесть месяцев. Для отдельных отраслей и организаций сроки проведения регулируются соответствующими отраслевыми и межотраслевыми нормативными правовыми актами по безопасности и охране труда;",

options: [

"не реже одного раза в шесть месяцев. Для отдельных отраслей и организаций сроки проведения регулируются соответствующими отраслевыми и межотраслевыми нормативными правовыми актами по безопасности и охране труда;",

"для работников, занятых на работах; с повышенной опасностью,ежеквартально, для остальных — ежегодно;",

"в соответствии с ответами «а» и «б».",

"Никогда"

]

},

{

numb: 13,

question: "Обязан ли работодатель обучать работников оказанию первой помощи пострадавшим?",

answer: "работодатель обязан организовать проведение периодического, не реже одного раза в год, обучения оказанию первой помощи пострадавшим.",

options: [

"да, при приеме на работу в соответствии с программой вводного инструктажа;",

"желательно;",

"работодатель обязан организовать проведение периодического, не реже одного раза в год, обучения оказанию первой помощи пострадавшим.",

"Не обязан"

]

},

# 4 ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

# 4.1 Организация используемой входной и выходной информации

Входные данные идентичны выходным.