

Лабораторная работа №9

Тема: " модульное программирование. AMD-концепции.

Оптимизация на клиентской стороне"

Преподаватель: Атрощенко Н.А.

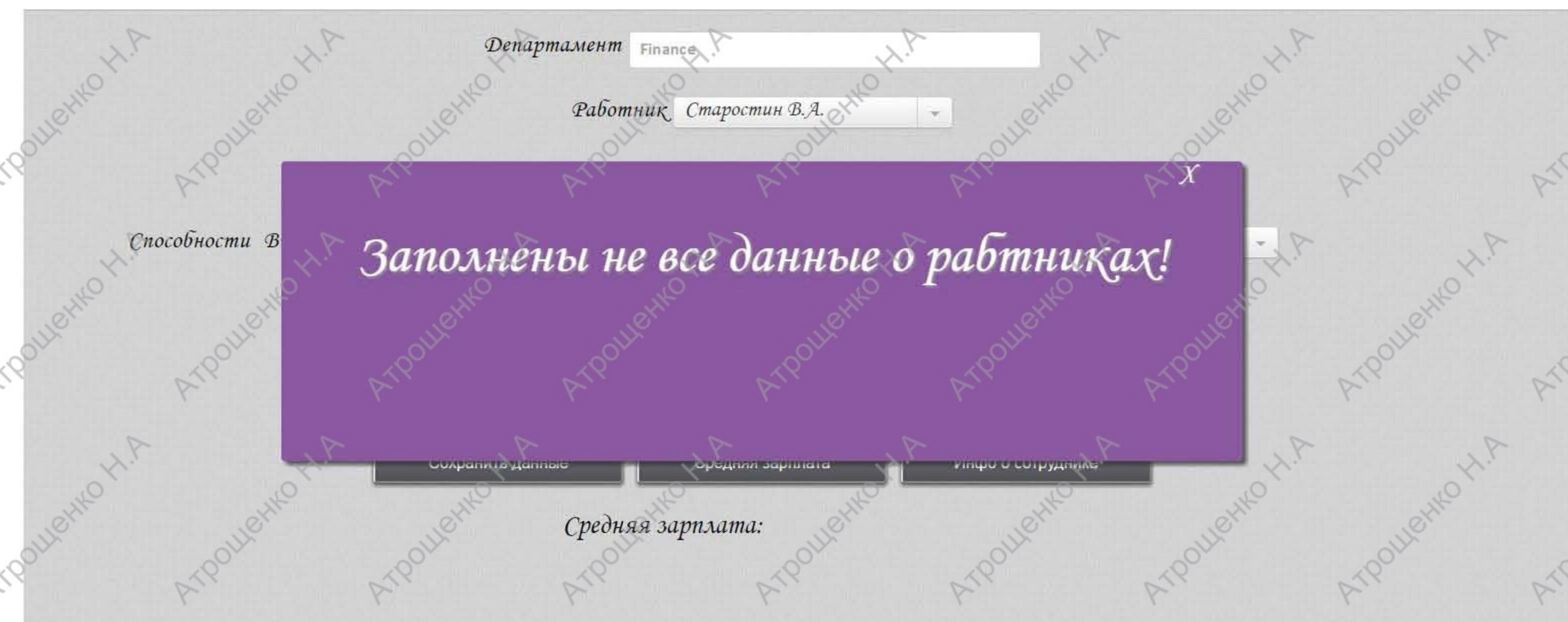
Цель: научиться создавать динамические веб-проекты с использованием и организацией модульного проектирования JavaScript.

Основные требования:

- Проект должен включать в себя папки: соответственно для вашего шаблона (с названием template.js) (шаблонов), css, html, js, json-файлов и resource и используемых библиотек файлов.
- Ключевой файл должен иметь название index.html.
- Рабочим языком для организации бизнес-логики проекта является JavaScript.
- Для организации вывода данных использовать возможности шаблонизатора JavaScript (рекомендуется mustache.js), запрещается использование других возможностей (других языков программирования, плагинов, фреймворков и т.д.).
- Подгрузка данных (например, фамилии сотрудников, нужные коэффициенты) должна осуществляться только из .json файлов.
- Для работы приложения использовать возможности модульной организации функций вычисления с помощью библиотеки require.js. В зависимости от выбранных опций пользователем на веб-странице должен подключаться нужный модуль, или группа модулей для расчёта итогового значения.
- Для хранения итоговой информации использовать возможности localStorage, sessionStorage как локального хранилища данных и подгружаться по нажатию на кнопку.
- Обязательна обработка всех исключительных ситуаций.
- В проекте должен быть единый стиль оформления и определённое тематическое оформление, поэтому картинки подбирайте соответственно с тематикой.

Образец выполнения:

The screenshot shows a web-based application interface. At the top, there are tabs for 'Департамент' (Department) set to 'Finance' and 'Работник' (Employee) set to 'Старостин В.А.'. Below these, there is a dropdown menu labeled 'Должность' (Position) which is currently open, showing a list of options: 'Механик' (Mechanic), 'Инженер' (Engineer), 'Программист' (Programmer), 'Экономист' (Economist), and 'Менеджер' (Manager). The background of the application shows some placeholder text and buttons for saving data and viewing employee info.

A screenshot of the same web application interface, but now with data entered. The 'Департамент' dropdown is empty. The 'Работник' dropdown shows 'Старостин В.А.'. The 'Способности' dropdown has options 'Высокие', 'Средние', and 'Низкие' with 'Средние' selected. The 'Вредность' dropdown shows 'Ночное время'. The 'Месячная тарифная ставка' input field contains '5'. The 'Удержание со счета' input field contains '56'. Below the form, a list of employee details is displayed:

- Имя рабочего: Старостин В.А.
- Профессия: Механик
- Способности: 3
- Вредность работы: 3
- Ставка: 5
- Удержание: 56
- Зарплата: 18000000

At the bottom are three buttons: 'Сохранить данные' (Save data), 'Средняя зарплата' (Average salary), and 'Инфо о сотруднике' (Employee info).

Индивидуальные задания:

1. Создать модульное приложение с использованием RequireJS, в котором рассчитывается и выводится средняя заработка 4 сотрудников отдела с разной тарифной ставкой по результатам вычислений, произведённых в 4 разных модулях, где учитывается : повышение за характер и специфику труда, сложность выполняемых работ, разряд, ставка с учётом понижения (госпошлина). Все необходимые коэф-ты, в том числе и тарифная ставка задаются пользователем (используйте т/поле, checkbox, выпад. список). Имена, департамент, должность сотрудников должны считываться с json-файлов. Разместите 2 кнопки, по нажатию на первую выводится средняя зарплата по отделу, а на вторую выводятся в виде нумерованного списка все данные о сотруднике, а также его зарплата.

2. Создать модульное приложение с использованием RequireJS, в котором рассчитывается и выводится средняя заработка 5 сотрудников отдела с разной тарифной ставкой по результатам вычислений, произведённых в 3 разных модулях, где учитывается : повышение за стаж работы в отрасли, коррект. коэф., ставка с учётом понижения (больничный лист). Все необходимые коэф-ты, в том числе и тарифная ставка задаются пользователем (используйте т/поле, checkbox, выпад. список). Имена, департамент, должность сотрудников должны считываться с json-файлов. Разместите 2 кнопки, по нажатию на первую выводится средняя зарплата по отделу, а на вторую выводятся в виде таблицы все данные о сотруднике, а также его зарплата.

3. Создать модульное приложение с использованием RequireJS, в котором рассчитывается и выводится средняя заработка 4 сотрудников отдела с разной тарифной ставкой по результатам вычислений, произведённых в 4 разных модулях, где учитывается : повышение за выслугу лет, надбавка за проф. мастерство, сложность выполняемых работ, ставка с учётом понижения (кредит). Все необходимые коэф-ты, в том числе и тарифная ставка задаются пользователем (используйте т/поле, checkbox, выпад. список). Имена, департамент, должность сотрудников должны считываться с json-файлов. Разместите 2 кнопки, по нажатию на первую выводится средняя зарплата по отделу, а на вторую выводятся в виде нумерованного списка все данные о сотруднике, а также его зарплата.

4. Создать модульное приложение с использованием RequireJS, в котором рассчитывается и выводится средняя заработка 5 сотрудников отдела с разной тарифной ставкой по результатам вычислений, произведённых в 3 разных модулях, где учитывается : повышение за переработку рабочего времени, за характер и специфику труда, за степень занятости выполнением работ. Все необходимые коэф-ты, в том числе и тарифная ставка задаются пользователем (используйте т/поле, checkbox, выпад. список). Имена, департамент, должность сотрудников должны считываться с json-файлов. Разместите 2 кнопки, по нажатию на первую выводится средняя зарплата по отделу, а на вторую выводятся в виде таблицы все данные о сотруднике, а также его зарплата.

5. Создать модульное приложение с использованием RequireJS, в котором рассчитывается и выводится средняя заработка 4 сотрудников отдела с разной тарифной ставкой по результатам вычислений, произведённых в 4 разных модулях, где учитывается : коэффициент повышения, учитывающего сложность выполняемых работ, надбавка за классность, за профессиональное мастерство, ставка с учётом понижения (кредит за учебу в вузе). Все необходимые коэф-ты, в том числе и тарифная ставка задаются пользователем (используйте т/поле, checkbox, выпад. список). Имена, департамент, должность сотрудников должны считываться с json-файлов. Разместите 2 кнопки, по нажатию на первую выводится средняя зарплата по отделу, а на вторую выводятся в виде нумерованного списка все данные о сотруднике, а также его зарплата.

6. Создать модульное приложение с использованием RequireJS, в котором рассчитывается и выводится средняя заработка 5 сотрудников отдела с разной тарифной ставкой по результатам вычислений, произведённых в 3 разных модулях, где учитывается : наличие пенсии, повышение за переработку рабочего времени, ставка с учетом понижения – брак в работе. Все необходимые коэф-ты, в том числе и тарифная ставка задаются пользователем (используйте т/поле, checkbox, выпад. список). Имена, департамент, должность сотрудников должны считываться с json-файлов. Разместите 2 кнопки, по нажатию на первую выводится средняя зарплата по отделу, а на вторую выводятся в виде таблицы все данные о сотруднике, а также его зарплата.

7. Создать модульное приложение с использованием RequireJS, в котором рассчитывается и выводится средняя заработка 5 сотрудников отдела с разной тарифной ставкой по результатам вычислений, произведённых в 3 разных модулях, где учитывается : повышение, за рационализаторские разработки, за вредность, ставка с учётом понижения (удержание по суду). Все необходимые коэф-ты, в том числе и тарифная ставка задаются пользователем (используйте т/поле, checkbox, выпад. список). Имена, департамент, должность сотрудников должны считываться с json-файлов. Разместите 2 кнопки, по

нажатию на первую выводится средняя зарплата по отделу, а на вторую выводятся в виде нумерованного списка все данные о сотруднике, а также его зарплата.

8. Создать модульное приложение с использованием RequireJS, в котором рассчитывается и выводится средняя заработка 4 сотрудников отдела с разной тарифной ставкой по результатам вычислений, произведённых в 4 разных модулях, где учитывается : повышение за экономию электроэнергии, северная надбавка, разряд, ставка с учётом понижения (налог на бездетность). Все необходимые коэф-ты, в том числе и тарифная ставка задаются пользователем (используйте т/поле, checkbox, выпад. список). Имена, департамент, должность сотрудников должны считываться с json-файлов . Разместите 2 кнопки, по нажатию на первую выводится средняя зарплата по отделу, а на вторую выводятся в виде таблицы все данные о сотруднике, а также его зарплата.

9. Создать модульное приложение с использованием RequireJS, в котором рассчитывается и выводится средняя заработка 5 сотрудников отдела с разной тарифной ставкой по результатам вычислений, произведённых в 3 разных модулях, где учитывается : повышение за классность, разряд, ставка с учётом понижения (проценты за несвоевременные выплаты). Все необходимые коэф-ты, в том числе и тарифная ставка задаются пользователем (используйте т/поле, checkbox, выпад. список). Имена, департамент, должность сотрудников должны считываться с json-файлов . Разместите 2 кнопки, по нажатию на первую выводится средняя зарплата по отделу, а на вторую выводятся в виде нумерованного списка все данные о сотруднике, а также его зарплата.

10. Создать модульное приложение с использованием RequireJS, в котором рассчитывается и выводится средняя заработка 4 сотрудников отдела с разной тарифной ставкой по результатам вычислений, произведённых в 4 разных модулях, где учитывается : повышение за экономию ден.ср-в, персон. надбавка, сложность выполняемых работ, ставка с учётом понижения (брак в работе). Все необходимые коэф-ты, в том числе и тарифная ставка задаются пользователем (используйте т/поле, checkbox, выпад. список). Имена, департамент, должность сотрудников должны считываться с json-файлов . Разместите 2 кнопки, по нажатию на первую выводится средняя зарплата по отделу, а на вторую выводятся в виде таблицы все данные о сотруднике, а также его зарплата.

11. Создать модульное приложение с использованием RequireJS, в котором рассчитывается и выводится средняя заработка 5 сотрудников отдела с разной тарифной ставкой по результатам вычислений, произведённых в 3 разных модулях, где учитывается : повышение за сложность выполняемых работ, за ночную смену, ставка с учётом понижения (кредит). Все необходимые коэф-ты, в том числе и тарифная ставка задаются пользователем (используйте т/поле, checkbox, выпад. список). Имена, департамент, должность сотрудников должны считываться с json-файлов . Разместите 2 кнопки, по нажатию на первую выводится средняя зарплата по отделу, а на вторую выводятся в виде нумерованного списка все данные о сотруднике, а также его зарплата.

12. Создать модульное приложение с использованием RequireJS, в котором рассчитывается и выводится средняя заработка 4 сотрудников отдела с разной тарифной ставкой по результатам вычислений, произведённых в 4 разных модулях, где учитывается : коэффициент повышения, учитывающего переработку рабочего времени, надбавку за стаж, за переработку рабочего времени, ставка с учётом понижения (исполнительный лист). Все необходимые коэф-ты, в том числе и тарифная ставка задаются пользователем (используйте т/поле, checkbox, выпад. список). Имена, департамент, должность сотрудников должны считываться с json-файлов . Разместите 2 кнопки, по нажатию на

первую выводится средняя зарплата по отделу, а на вторую выводятся в виде таблицы все данные о сотруднике, а также его зарплата.

13. Создать модульное приложение с использованием RequireJS, в котором рассчитывается и выводится средняя заработка 5 сотрудников отдела с разной тарифной ставкой по результатам вычислений, произведённых в 3 разных модулях, где учитывается: наличие персональной пенсии, повышение за стаж, ставка с учетом понижения – нарушение техники безопасности. Все необходимые коэф-ты, в том числе и тарифная ставка задаются пользователем (используйте т/поле, checkbox, выпад. список). Имена, департамент, должность сотрудников должны считываться с json-файлов. Разместите 2 кнопки, по нажатию на первую выводится средняя зарплата по отделу, а на вторую выводятся в виде нумерованного списка все данные о сотруднике, а также его зарплата.

14. Создать модульное приложение с использованием RequireJS, в котором рассчитывается и выводится средняя заработка 5 сотрудников отдела с разной тарифной ставкой по результатам вычислений, произведённых в 3 разных модулях, где учитывается: повышение, за профессиональную категорию, за переработку раб. времени, ставка с учётом понижения (налог на доходы). Все необходимые коэф-ты, в том числе и тарифная ставка задаются пользователем (используйте т/поле, checkbox, выпад. список). Имена, департамент, должность сотрудников должны считываться с json-файлов. Разместите 2 кнопки, по нажатию на первую выводится средняя зарплата по отделу, а на вторую выводятся в виде таблицы все данные о сотруднике, а также его зарплата.