

Написать шаблон компонента-клиента-исследователя, используя средства HTML и JavaScript.

Общее описание

Разрабатываемый шаблон необходим как тонкий клиент, обеспечивающий взаимодействие пользователя с толстым клиентом (виртуальным интерфейсом). Удаленный виртуальный интерфейс, в свою очередь, взаимодействует с сервером. Общение между двумя клиентами осуществляется через POST запросы.

В качестве типа графического интерфейса используется окно, разделенное на несколько рабочих областей — панель управления, древо проектов, основная рабочая область, информационная панель. Древо проектов содержит иерархическую структуру проектов пользователя в корне которой находятся проекты пользователя, содержащие скрипты, которые, в свою очередь, содержат снимки старых версий. Основная область содержит несколько вкладок, число которых равно количеству скриптов в текущем проекте плюс вкладка результатов моделирования. Вкладка результатов моделирования содержит список результатов моделирования и область со значениями выбранного результата в табличной форме. Информационная панель содержит две вкладки — вкладка с логом и вкладка с графиками.

Порядок и варианты взаимодействия перечисленных элементов интерфейса с пользователем приведены на рисунке 1. Основной принцип, заложенный в представленную диаграмму состояния — любое действие пользователя заканчивается отправкой виртуальному интерфейсу сообщения с префиксом «esc» в соответствии с описанием, приведенным в таблице 1. После завершения обработки сообщения, ядро клиента меняет виртуальный интерфейс, а графический интерфейс «узнает» об этом в момент синхронизации с виртуальным на стороне сервера (периодическое сообщение Timer на рисунке 1).

Внешний вид графического интерфейса представлен на рисунке 2.

Внимание: разработать нужно **только шаблон (см. Рисунок 2)**, в котором:

1. Есть функционирующий интерфейс, оформленный аккуратно с использованием стилей;
2. Цветовая палитра интерфейса: 0x535251, 0x808080, 0xE4E2E0, 0x6B6B6B, 0xFFFFFFFF.
3. Оформление и реакция интерфейса на действия пользователя (перемещение мыши, нажатие на кнопки) должны быть максимально близки к стилю Qt5 fusion. *Иконки в формате png будут переданы по запросу;*
4. Есть диалоговое окно авторизации, которое появляется при нажатии кнопки 1.1. В окне реализовать ввод имени пользователя и пароля, ничего лишнего;
5. Реализовано древо проектов. Корневые элементы — проекты. Внутри корневых элементов расположены скрипты. Внутри скриптов — снимки. Каждый элемент дерева имеет иконку и наименование. В древе проектов всегда активен только один проект и один скрипт со списком снимков, остальные проекты и скрипты свернуты. Реализовать в виде отдельных подпрограмм JS обновление древа проектов (3 подпрограммы — обновление списка проектов, списка скриптов в проекте, списка снимков в скрипте) на основе переданного в

подпрограмму объекта-строки. Объект-строка содержит список наименований, разделенных символами разрыва строки и переноса каретки (#13#10). С обновлением списка скриптов должен меняться перечень вкладок в области 4.

6. В древе проектов должна быть реализована возможность изменять имя проекта и скрипта.

7. При нажатии кнопки 2.1 должно возникать выпадающее меню с элементами: Создать проект, Создать скрипт, Ассоциировать скрипт. Должны быть реализованы обработчики нажатия на указанные элементы меню. При нажатии «Создать проект» и «создать скрипт» необходимо вывести модальное окно с запросом имени нового скрипта/проекта. В обработчики подтверждения выбора пользователя должны быть вставлены пустые POST запросы;

8. Должны быть реализованы обработчики нажатия на кнопки 1.2, 1.3 и 2.2 и двойные щелчки по элементам древа проектов. В обработчики должны быть вставлены пустые POST запросы;

9. Должна быть реализована возможность перехода по вкладкам (6.1 и 6.2);

10. Вводить текст в основную рабочую область 5 (подсветку синтаксиса реализовывать не нужно);

11. Вкладку результатов 4.1 реализовывать не нужно;

12. Вкладку списка графиков реализовывать не нужно, но при этом обозначить ее списком элементов от 1 до 10 (типа ListBox);

13. Вкладку лога реализовать в виде текстового поля. Реализовать в виде подпрограммы JS обновления лога путем полного обновления содержимого указанного текстового поля.

14. Реализовать периодический (каждую секунду) пустой POST запрос, запускаемый после загрузки графического интерфейса.

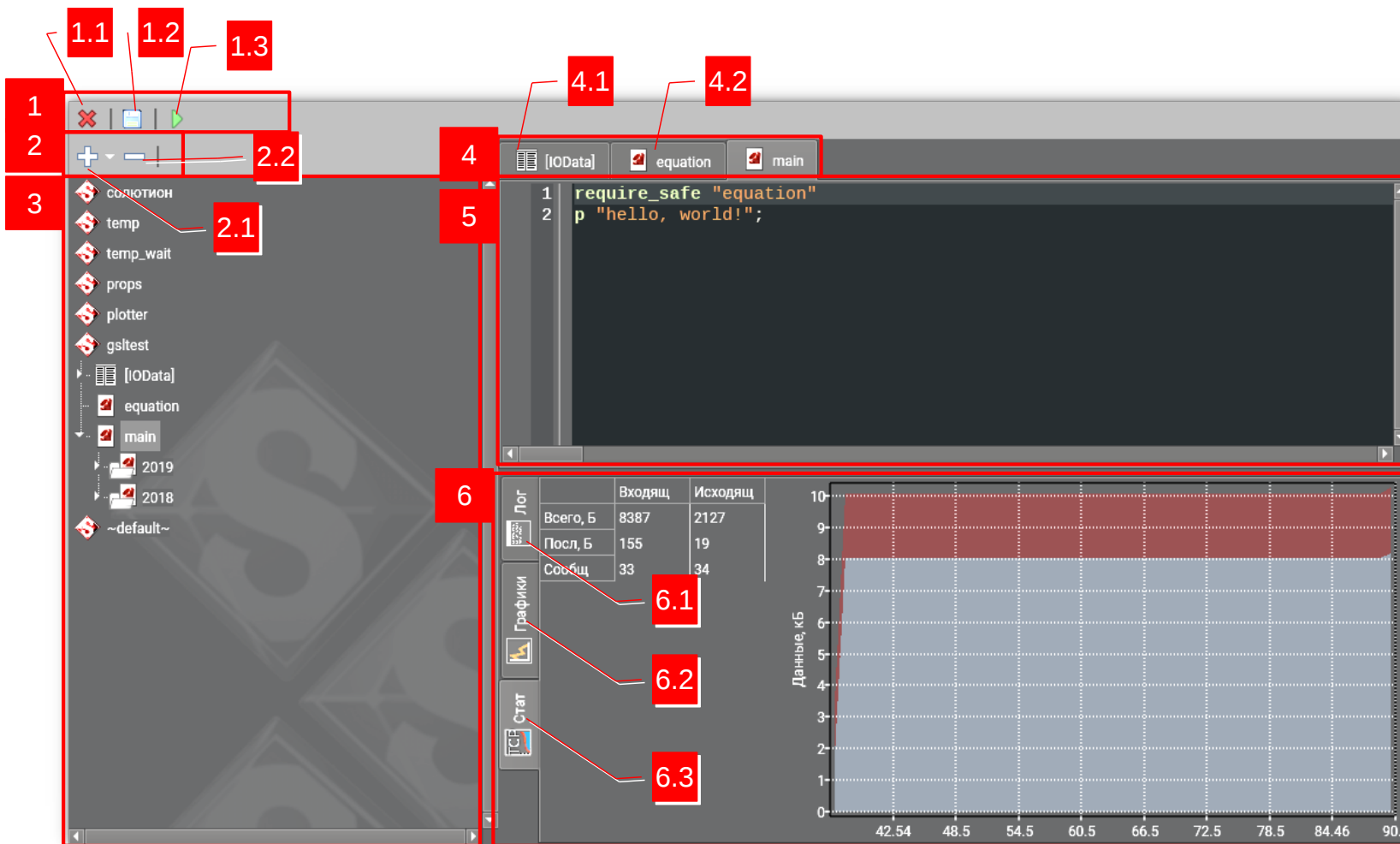


Рисунок 2 — Иллюстрация графического интерфейса пользователя компонента-клиента-исследователя для настольных компьютеров. 1 — панель инструментов, 2 — панель инструментов древа проектов, 3 — древо проектов, 4 — вкладки рабочей области, 5 — рабочая область (текстовый редактор скрипта), 6 — информационная область. 1.1 — Подключить/закрыть подключение, 1.2 — сохранить скрипт, 1.3 — запустить скрипт, 2.1 — кнопка «Создать», 2.2 — кнопка «Удалить», 4.1 — вкладка «Результаты», 4.2 — вкладки скриптов, 6.1 — вкладка «Лог», 6.2 — вкладка «Графики», 6.3 — вкладка «Статистика» (не реализовывать!)

№	Наименование	Параметры исходящего запроса	Ответ виртуального интерфейса	Описание