

Базовой технологией WWW является протокол передачи данных под названием HyperText Transfer Protocol (протокол передачи гипертекста), сокращенно HTTP. HTTP протокол является одной из основополагающих Web. Термин Web был определен Тимом Бернерс-Ли [1] и состоял из трех основных составляющих:

- URI (UniformResourceIdentifier) – путь к уникальному ресурсу в сети.
- HTML (HyperTextMarkupLanguage) – стандартизированный язык разметки.
- HTTP (HyperTextTransferProtocol) – протокол передачи данных в сети интернет.

Работа с документами Web-сервера осуществляется при помощи браузеров, таких как Firefox, Mozilla, IE и другие. При этом браузер становится клиентом по отношению к серверу, формирует запросы в соответствии с правилами и отправляет их серверу. Инициатором запроса всегда является клиент. Сервер может обращаться к другим серверам в процессе обработки запроса. Сервер может отослать свою или чужую страницу HTML, адрес которой запросил клиент.

Стандартизированным языком, при помощи которого происходит общение между клиентом и сервером, является XML (eXtensible Markup Language) – расширяемый язык разметки. Для коммуникации посредством протокола HTTP используется ряд методов [2]. Наиболее используемые из них: GET, POST, PUT, DELETE. Метод GET используется для получения информации или какого-либо ресурса. Взаимодействие при помощи метода POST аналогично использованию метода GET с тем отличием, что при помощи метода POST можно передавать какие-то данные на сервер, т.е. добавляется еще один параметр: «Тело» запроса с данными, передаваемыми на сервер. Метод PUT аналогичен методу POST. При этом метод PUT используется для добавления новых ресурсов на сервер, а метод POST используется для обновления уже существующих данных на сервере. Метод DELETE используется для удаления каких-либо данных с сервера. При использовании любого из методов используется URL (единый указатель ресурса).

При отправке данных на сервер любым методом передаются не только сами данные, введенные пользователем, но и ряд переменных, называемых переменными окружения, характеризующими клиента, историю его работы, пути к файлам и т. п.: REMOTE\_ADDR (IP-адрес компьютера, отправляющего запрос), REMOTE\_HOST (имя компьютера, с которого отправлен запрос), HTTP\_REFERER (адрес страницы, ссылающейся на текущий скрипт), REQUEST\_METHOD (метод, который был использован при отправке запроса), QUERY\_STRING (информация, находящаяся в URL после знака вопроса), SCRIPT\_NAME (виртуальный путь к программе, которая должна выполняться), HTTP\_USER\_AGENT (информация о браузере, который использует клиент), CONTENT\_TYPE (при использовании метода передачи POST в этой переменной хранится строка с названием типа данных, присланных браузером).

Список использованных источников:

1. Википедия [Электронный ресурс]. – Электронные данные. – Режим доступа: [https://ru.wikipedia.org/wiki/Всемирная\\_паутина](https://ru.wikipedia.org/wiki/Всемирная_паутина).
2. Общий принцип работы Интернет приложений [Электронный ресурс]. – Электронные данные. – Режим доступа: [http://eos.ibi.spb.ru/umk/5\\_15/5/5\\_R1\\_T1.html](http://eos.ibi.spb.ru/umk/5_15/5/5_R1_T1.html).

## **ВЕБ-ПРИЛОЖЕНИЕ ПО СОЗДАНИЮ И УПРАВЛЕНИЮ ПЛАНОМ ХИМИЧЕСКИХ ЭКСПЕРИМЕНТОВ ДЛЯ РОБОТИЗИРОВАННЫХ СТАНЦИЙ**

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Шевцов Д.О., Шевцова Т.А.*

*Куликов С.С. – к.т.н., доцент*

Разработано веб-ориентированное приложение, написанное на языке Java с использованием Java-фреймворка GoogleWebToolkit. Для реализации пользовательского интерфейса использовали библиотеку Sencha. В качестве локальной базы использовали MongoDB, а централизованной – OracleDB.

Актуальной задачей современной аналитической химии является решение проблем автоматизации методов химического анализа. Возрастающая потребность в полной или, по крайней мере, частичной механизации и автоматизации аналитического процесса напрямую связана с постоянно растущей необходимостью выполнения огромного числа рутинных анализов в самых разных сферах деятельности человека – медицине, фармацевтике, службах контроля окружающей среды, криминалистике и др. [1].

Благодаря непрерывному развитию техники появляются все новые специализированные устройства, способные заменить человека на всех стадиях химического анализа. Важным при автоматизации аналитического процесса и последующем его управлении является установление связи между всеми стадиями анализа, а также между химиком и техническим устройством, которое осуществляется посредством компьютерных программ.

Разработанное приложение призвано решить проблему интеграции современной автоматизированной техники в химические лаборатории и ориентировано на связь между химиком и робототехникой. Подобные

приложения призваны стать удобной платформой, позволяющей легко конфигурировать и проводить химические эксперименты.

Разработанное приложение реализовано при помощи языка Java с использованием Java-фреймворка GoogleWebToolkit. Преимущество веб приложения – независимость от клиентской архитектуры: для работы с приложением достаточно иметь устройство с браузером и доступ в сеть. Сконфигурированные эксперименты, содержащие в себе информацию о разработке экспериментальных методик, подборе условий анализа, а также результаты эксперимента, хранятся централизованно, способствуя возможности интеграции различных лабораторий в единый исследовательский центр.

При создании приложения решался ряд задач, а именно: реализация веб-интерфейса для создания плана химического эксперимента; возможность сохранения и загрузки созданного пользователем химического эксперимента; возможность экспорта в настольное приложение для последующего проведения эксперимента автоматическими роботизированными станциями; возможность импорта эксперимента из настольного приложения с возможностью представления результатов в удобном виде.

В качестве фреймворка использовали GoogleWebToolkit (GWT), который позволяет создавать Ajax-приложения. Основными преимуществами GWT для разработки данного приложения являются встроенный компилятор из Java в JavaScript, простой механизм удалённого вызова процедур (RPC), динамические и многоразовые компоненты пользовательского интерфейса (виджеты), заранее разработанные классы (drag'n'drop и др.), ряд доступных от третьих лиц библиотек (от Google и др.).

В качестве одной из таких библиотек-расширений использовали библиотеку Sencha – это набор стандартных виджетов, панелей и др. инструментов для представления пользовательского интерфейса.

В качестве локального хранилища промежуточных версий экспериментов использована нереляционная база данных MongoDB. На финальной стадии планирования эксперимента (непосредственного перед его выполнением) данные экспортируются в настольное приложение, где и происходит его выполнение. Для хранения финальных версий экспериментов использовали базу данных Oracle DB.

Работа с приложением начинается с авторизации, после её успешного прохождения предоставляется выбор между созданием, загрузкой или копированием существующего эксперимента. При создании нового эксперимента необходимо выбрать рабочее пространство – микропланшет с определённым количеством лунок и объёмом в каждой из них.

Создание плана проведения химического эксперимента происходит в виртуальных микропланшетах. Вкладка «DefineRecipes» содержит выбранные виртуальные микропланшеты с лунками, в которые можно добавлять различные растворы и вещества. Анализ и операции могут быть применены либо ко всему виртуальному микропланшету, например, перемешивание или нагревание, либо к отдельным лункам, например, флуоресцентный анализ. Рисунок 1 демонстрирует добавление терефталевой кислоты к диапазону лунок A1-H1, в качестве способа заполнения выбран градиентный с шагом 10 мкл.

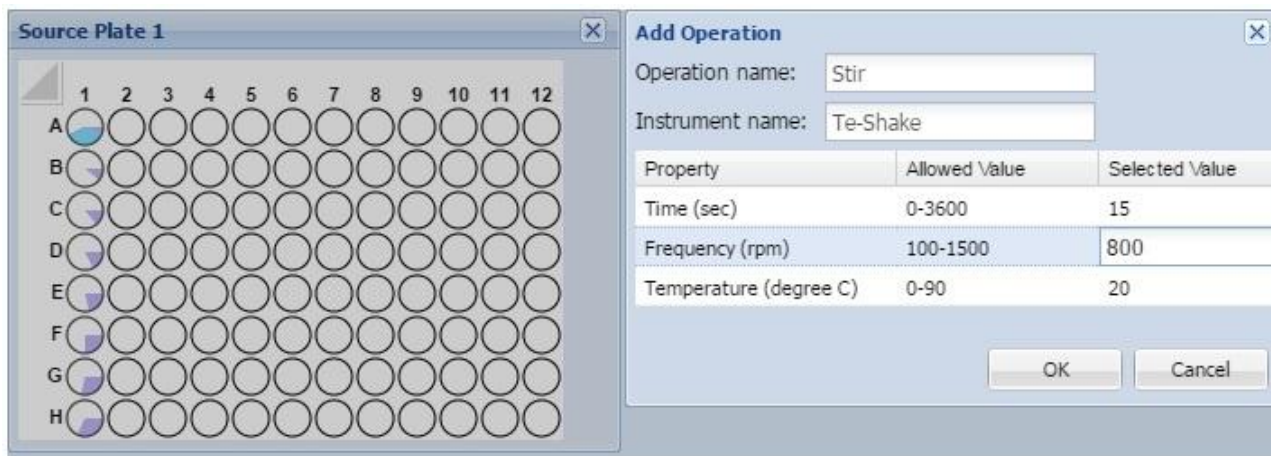


Рисунок 1. Микропланшет и добавление к нему операции «перемешивание» с выбором устройства и параметров проведения.

Вкладка «ExperimentOverview» предоставляет возможность провести обзор эксперимента (рисунок 2).

Experiment Data		Select Materials		Define Mixtures		Select Operations		Define Recipe		Define Analysis		Experiment Overview		Instruction List	
Plate Name	Row	Column	Action	Class	Action Content		Amount	Units							
Plate 1	A	1	Add	Reagent	C8H6O4 (Terephthalic acid)		20	µl							
Plate 1	A	1	Add		FeSO4 0.05M		100	µl							
Plate 1	A	1	Operation		Stir										
Plate 1	B	1	Add	Reagent	C8H6O4 (Terephthalic acid)		30	µl							
Plate 1	C	1	Add	Reagent	C8H6O4 (Terephthalic acid)		40	µl							
Plate 1	D	1	Add	Reagent	C8H6O4 (Terephthalic acid)		50	µl							
Plate 1	E	1	Add	Reagent	C8H6O4 (Terephthalic acid)		60	µl							
Plate 1	F	1	Add	Reagent	C8H6O4 (Terephthalic acid)		70	µl							
Plate 1	G	1	Add	Reagent	C8H6O4 (Terephthalic acid)		80	µl							
Plate 1	H	1	Add	Reagent	C8H6O4 (Terephthalic acid)		90	µl							

Рисунок 2. Вкладка «ExperimentOverview»

Последняя вкладка в приложении – «InstructionList» – содержит список последовательных инструкций, которые будут использованы роботами для получения результатов эксперимента. Она предназначена для заключительной проверки правильности созданного экспериментального плана.

После окончательной проверки полное описание и последовательность проведения операций эксперимента сохраняется и экспортируется в лабораторию, в которой он проводится в автоматическом режиме.

Список использованных источников:

1. Отто, М. Автоматизация анализа и производственный анализ / М. Отто // Современные методы аналитической химии. – М., 2008. – Гл. 7. – С. 436-478.
2. Yuan, L. Automation in new frontiers of bioanalysis: a key for quality and efficiency / Y. Long, J. Qin C // J. Bioanalysis. – 2012. – Vol. 4, № 23. – P. 2759-2762.
3. Dunkerley, S. A general robot control system for automatic chemical analysis / S. Dunkerley, M. J. Adams // Laboratory Automation and Information Management – 1997. – Vol. 33, № 2. – P. 93-105.
4. North, N. Robotics and Laboratory Automation in Pharmaceuticals Analysis / N. North // Encyclopedia of Analytical Chemistry. – 2006.

## ИСПОЛЬЗОВАНИЕ ШТРИХ-КОДОВ ДЛЯ АВТОМАТИЗАЦИИ РАБОТЫ С ТЕСТОВЫМИ ЗАДАНИЯМИ

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Шендерович В.А.*

*Деменковец Д.В. – ассистент кафедры ПОИТ*

Наряду с компьютерным тестированием, в учебном процессе сохраняет актуальность такая форма контроля знаний, при которой тестовые задания выдаются студентам в печатном виде. Разработанный программный продукт «Генератор тестов» предназначен для автоматизации подготовки тестовых заданий и обработки полученных ответов с применением штрих-кодов. Данное программное средство позволит снизить трудоемкость и повысить качество тестирования, а также систематизировать тестовые задания в базе данных вопросов и ответов.

Программное средство «Генератор тестов» реализует основные функции, необходимые для формирования базы данных вопросов и ответов по учебному предмету с разбивкой по темам и подготовки на этой основе тестовых заданий. Перед формированием тестов пользователь программы осуществляет ввод в базу данных вопросов и ответов. После заполнения базы появляется возможность подготовки, редактирования, сохранения и распечатки тестовых заданий, создаваемых путем случайной выборки вопросов по определенной теме с заданными преподавателем параметрами. В частности, могут варьироваться: число вариантов, количество и сложность вопросов, а также повторяемость вопросов в различных вариантах.

На данном этапе разработки «Генератор тестов» обеспечивает следующие функции:

- добавление в базу новых тем, переименование и удаление существующих;
- добавление в существующие темы новых вопросов, редактирование и удаление существующих вопросов;
- добавление к существующим вопросам новых ответов, редактирование и удаление существующих ответов;
- создание и сохранение случайно сгенерированных тематических тестов с заданием количества вариантов в тесте и количества вопросов по каждому уровню сложности, а также удаление уже сохранённых тестов;