



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»

ШКОЛА ЕСТЕСТВЕННЫХ НАУК

Кафедра компьютерных систем

Шевцов Роман Александрович

**РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА ДЛЯ СИСТЕМЫ
АВТОМАТИЧЕСКИХ МЕХАНИЧЕСКИХ ИСПЫТАНИЙ**

БАКАЛАВРСКАЯ РАБОТА

по основной образовательной программе подготовки бакалавров
по направлению 09.03.02 – Информационные системы и технологии

г. Владивосток
2018

Аннотация

Выпускная квалификационная работа на тему: «Разработка программного комплекса для системы автоматизированных механических испытаний»

Автор: Шевцов Роман Александрович, студент 4 курса

Выпускная квалификационная работа 43 с., 3 ч., 20 рис., 13 источников

Ключевые слова: клиент-серверное взаимодействие, сокет, формат обмена данными, JSON, Java, Python, SQLite, JFreeChart

Цель работы – разработка программного комплекса для системы автоматизированных механических испытаний

Задачи:

- Изучение инструментов разработки программного комплекса;
- Поиск существующих и создание собственных архитектурных решений для разрабатываемой системы;
- Реализация программного комплекса.

Дипломная работа состоит из введения, трех глав и заключения.

В введении представлено обоснование выбора темы, раскрыта актуальность выбранной темы, сформулирована цель и поставлены задачи необходимые для достижения цели работы. В первой главе выполняется обзор средств разработки программного комплекса для системы автоматизированных механических испытаний.

В второй главе рассматриваются возможные архитектурные решения для разрабатываемого программного комплекса. В конце главы представлено техническое задание к проекту.

В третьей главе представлена реализация программного комплекса согласно сформулированному во второй главе техническому заданию.

Заключение содержит выводы, основанные на полученных результатах.

Оглавление

Введение	4
Глава 1. Обзор средств разработки программного комплекса для системы автоматизированных механических испытаний	6
1.1 Средства разработки серверного приложения.....	6
1.2 Средства разработки клиентского приложения	9
Глава 2. Обзор архитектурных решений	14
2.1 Модель клиент-серверного взаимодействия.....	14
2.2 Обзор сокетов	17
2.3 Формат обмена данными	18
2.4 Выбор СУБД.....	21
2.5 Требования заказчика	25
2.6 Выбор архитектурных решений	26
2.7 Техническое задание	27
Глава 3 Реализация программного комплекса.....	30
3.1 Серверное приложение	30
3.2 Клиентское приложение	35
Заключение	42
Список литературы	43

Введение

С появлением вычислительной техники появилась возможность автоматизировать и упростить различные виды деятельности, в том числе процесс механических испытаний. Системы решающие данную задачу могут обеспечить считывание, обработку, хранение и отображение в реальном времени данных о ходе испытания, а также управление приборами системы.

Для простоты разработки и удобства пользователя подобные системы строят на основе архитектуры клиент-сервер. Серверное приложение решает задачи считывания, обработки, хранения и передачи данных клиентскому приложению. Клиентское приложение решает задачи приема данных от сервера, отображения данных в виде понятному для пользователя, а также обеспечивает возможность редактирования данных, хранящихся на сервере.

Данный программный комплекс должен решать следующие задачи:

- Считывать с определенной частотой данные с датчиков с помощью API на серверном приложении
- Передавать считанные данные клиентскому приложению с минимальными задержками, а также записывать эти данные в базу данных
- На клиентском приложении отображать полученные данные в удобном для пользователя виде
- Иметь возможность отображать данные хранящиеся в базе данных на клиентском приложении.

Актуальность работы заключается в том, что разрабатываемая система гораздо дешевле аналогичных комплексов на рынке.

Цель дипломной работы: разработать программный комплекс системы автоматических механических испытаний.

Цель может быть достигнута с помощью решения следующих задач:

- Изучить инструменты разработки программного комплекса

- Найти существующие и создать собственные архитектурные решения для разрабатываемой системы
- Реализовать программный комплекс

Глава 1. Обзор средств разработки программного комплекса для системы автоматизированных механических испытаний

1.1 Средства разработки серверного приложения

Существует множество языков программирования, используемых для разработки серверных приложений. В техническом плане в большинстве случаев нет каких-либо ограничений при выборе языка, то есть практически любой функционал приложения может быть успешно реализован на любом из них. В данной курсовой работе языком программирования для серверного приложения будет выбран Python.

1.1.1 Python

Python – это один из наиболее популярных современных языков программирования. Он пригоден для решения разнообразных задач и предлагает те же возможности, что и другие языки программирования: динамичность, поддержку ООП и кроссплатформенность.

Любой язык, неважно – для программирования или общения, состоит как минимум из двух частей – словаря и синтаксиса. Язык Python организован точно так же, предоставляя синтаксис для формирования выражений, образующих исполняемые программы, и словарь – набор функциональности в виде стандартной библиотеки и подключаемых модулей.

Стандартные модули расширения Python – это отлично спроектированная и неоднократно проверенная функциональность для решения задач, возникающих в каждом проекте по разработке ПО, обработка строк и текстов, взаимодействие с операционной системой, поддержка Web-приложений. Эти модули также написаны на языке Python, поэтому обладают его важнейшим свойством – кроссплатформенностью, позволяющей безболезненно и быстро переносить проекты с одной операционной системы на другую [\[1\]](#).

1.1.1.1 Библиотека Socket

Для обеспечения связи между клиентским и серверным приложением используются сокеты. Сокет — название программного интерфейса для обеспечения обмена данными между процессами. Процессы при таком обмене могут исполняться как на одной ЭВМ, так и на различных ЭВМ, связанных между собой сетью. Сокет — абстрактный объект, представляющий конечную точку соединения [\[2\]](#).

В список стандартных модулей расширений Python входит библиотека `socket`. Модуль `socket` обеспечивает возможность работать с сокетами из Python. Сокеты используют транспортный уровень согласно семиуровневой модели OSI (Open Systems Interconnection, взаимодействие открытых систем).

В начале для серверного приложения необходимо создать сокет с помощью метода `socket.socket()`, в которую мы передаем два аргумента: первый аргумент говорит о том, что это интернет-сокет, второй — что мы используем TCP-протокол.

Далее с помощью метода `socket.bind()` инициализируем ip-адрес и порт. При этом проверяется, не занят ли порт другой программой.

С помощью функции `socket.listen()` устанавливаем количество клиентских соединений, которые будет обслуживать операционная система.

Функция `socket.accept()` блокирует приложение до тех пор, пока не придет сообщение от клиента. Функция возвращает кортеж из двух параметров — объект самого соединения и адрес клиента.

Функция `socket.recv()` — читает данные из сокета. Аргумент устанавливает максимальное количество байтов в сообщении.

Функция `socket.send()` — отсылает данные клиенту.

После передачи данных, сокет необходимо закрыть с помощью метода `socket.close()` [\[3\]](#).

1.1.1.2 JSON

Сокеты позволяют передавать между клиентом и сервером данные в виде массива байтов. Для передачи объектов необходима сериализация.

Сериализация — это процесс преобразования объекта в поток байтов для сохранения или передачи в память, в базу данных или в файл. Эта операция предназначена для того, чтобы сохранить состояния объекта для последующего воссоздания при необходимости. Обратный процесс называется десериализацией.

В список стандартных модулей расширений Python входит библиотека json, предоставляющее API для сериализации объектов в формате JSON.

JSON (JavaScript Object Notation), определённый в стандарте RFC 4627, является простым форматом обмена данными, основанным на подмножестве синтаксиса JavaScript syntax (ECMA-262 3rd edition).

Для сериализации объекта с помощью модуля json необходимо импортировать модуль и вызвать метод json.dumps(), передав в качестве аргумента сериализуемый объект [4].

1.1.1.3 Python DB-API

Python DB-API – это не конкретная библиотека, а набор правил, которым подчиняются отдельные модули, реализующие работу с конкретными базами данных. Отдельные нюансы реализации для разных баз могут отличаться, но общие принципы позволяют использовать один и тот же подход при работе с разными базами данных.

Python имеет встроенную поддержку многих баз данных, для этого необходимо в скрипте указать импорт выбранной библиотеки <название библиотеки>.

Далее необходимо установить соединение (conn) с базой данных с помощью метода <название библиотеки>.connect().

Для дальнейшей работы с базой с помощью метода `conn.cursor()` создается курсор (`cursor`) – это специальный объект, который делает запросы и получает результаты.

Для формирования запроса к базе вызывается метод `cursor.execute()`, который в качестве аргумента принимает строку запроса в обычном SQL-синтаксисе. А для получения результата запрос – `cursor.fetchall()` [5].

1.2 Средства разработки клиентского приложения

1.2.1 Java

Java — сильно типизированный объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle).

Программы на Java транслируются в байт-код Java, выполняемый виртуальной машиной Java (JVM) — программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор. По скорости исполнения однотипные программы на java уступают в 1,5-2 раза программам на C/C++, при этом превосходят JavaScript, Ruby, Python.

Надёжность обеспечивается двумя принципами:

- ООП. Иерархия наследования увеличивает читаемость кода и снижает количество невынужденных ошибок.
- Строгая типизация. Разработчику приходится выполнять больший объём работы, но данные интерпретируются однозначно.

Синтаксис языка был унаследован от C++. Сегодня на фоне Python, Groove или Go его трудно назвать простым, однако тогда эволюционный вид позволил привлечь внимание Си-разработчиков.

Кроме сохранения общей формы конструкций, Java по сравнению с C++ формально лишился двух потенциальных опасностей: указателей и множественного наследования. На деле обе функции сохранены, но представлены в ином виде: вместо указателей используются значения, а в множественном наследовании участвуют не классы, а интерфейсы. Тем не менее,

такая особенность java программирования почти исключает возможный урон от невнимательности разработчика [6].

1.2.1.1 Socket

Ключевыми классами для реализации взаимодействия программ по TCP/UDP являются `java.net.Socket` и `java.net.ServerSocket`.

`java.net.Socket`- этот класс реализует клиентские сокеты.

В начале, клиентское приложение запускает подключение сокета по ip-адресу и порту сокета серверного приложения с помощью конструктора класса `Socket`.

Проверить состояние канала можно с помощью метода `socket.isOutputShutdown()`.

Для передачи данных на сервер, используется поток вывода, который можно получить вызвав метод `socket.getOutputStream()`. Для удобства работы с потоком вывода, который передает данные в виде массива байтов, его можно обернуть в поток `PrintWriter`, позволяющий записывать в поток текстовые данные.

Для приема данных с сервера, используется поток ввода, который можно получить вызвав метод `socket.getInputStream()`. Для удобства работы с потоком ввода, который принимает данные в виде массива байтов, его можно обернуть в поток `BufferedReader`, позволяющий считывать текстовые данные [7].

1.2.1.2 JSON

Клиентское приложение принимает от серверного данные в виде объектов в формате JSON, которые необходимо десериализовать. Стандартные библиотеки Java не поддерживают работу с JSON.

`Gson` – это сторонняя библиотека Java, которая может быть использована для преобразования объектов Java в их представление JSON. Его также можно использовать для преобразования JSON в эквивалентный объект Java.

Цели `Gson`:

- Обеспечьте простые в использовании механизмы, такие как `toString()` и конструктор (фабричный метод) для преобразования Java в JSON и наоборот
- Разрешить преобразование ранее не модифицируемых объектов в JSON и обратно
- Разрешить пользовательские представления для объектов
- Поддержка произвольно сложных объектов
- Создание компактного и легко читаемого выхода JSON

Для работы с Gson необходимо скачать и подключить библиотеку.

Следующим шагом нужно инициализировать объект класса Gson с помощью конструктора.

Чтобы десериализовать строку JSON, ее необходимо передать в качестве первого аргумента в метода `gson.fromJson()`, а в качестве второго аргумента необходимо использовать класс, к которому будет принадлежать десериализованный объект [\[8\]](#).

1.2.1.3 Swing

На сегодняшний день Swing остается самым популярным фреймворком для создания пользовательских интерфейсов на Java.

Компоненты Swing полностью написаны на Java. Для отрисовки используется 2D, что принесло с собой сразу несколько преимуществ. Набор стандартных компонентов значительно превосходит AWT по разнообразию и функциональности. Стало легко создавать новые компоненты, наследуясь от существующих и рисуя все, что душе угодно. Стала возможной поддержка различных стилей и скинов. Вместе с тем скорость работы первых версий Swing оставляла желать лучшего.

Тем не менее благодаря простоте использования, богатой документации и гибкости компонентов Swing стал, пожалуй, самым популярным графическим фреймворком в Java. На его базе появилось много расширений, таких как SwingX, JGoodies, которые значительно упрощают создание сложных

пользовательских интерфейсов. Практически все популярные среды программирования Java включают графические редакторы для Swing-форм.

Достоинства:

- часть JDK, не нужно ставить дополнительных библиотек
- по Swing большое количество книжек и ответов на форумах
- встроенный редактор форм почти во всех средах разработки
- на базе Swing есть много расширений типа SwingX
- поддержка различных стилей (Look and feel)

Недостатки:

- окно с множеством компонентов начинает подтормаживать;
- работа с менеджерами компоновки может стать настоящим кошмаром в сложных интерфейсах.

Для создания графического интерфейса пользователя, необходимо подключить библиотеки Swing.

Далее инициализировать окно, создав объект класса JFrame.

Следующим этапом настраиваются основные параметры окна и добавляются элементы интерфейса.

Чтобы пользователь мог увидеть интерфейс, необходимо вызвать метод setVisible() у объекта JFrame с аргументом true [\[9\]](#).

1.2.2 JFreeChart

JFreeChart – полностью бесплатная графическая Java библиотека, которая позволяет разработчикам легко отображать графики высокого качества в своих приложениях.

JFreeChart включает в себя:

- согласованный и хорошо документированный API, поддерживающий широкий диапазон типов диаграмм;
- гибкий дизайн, который легко расширяется и ориентирован как на серверные, так и на клиентские приложения;

- поддержка многих типов вывода, включая компоненты Swing и JavaFX, файлы изображений (включая PNG и JPEG) и форматы векторных графических файлов (включая PDF, EPS и SVG);
- JFreeChart - это библиотека с открытым исходным кодом. Она распространяется в соответствии с лицензией GNU Lesser General Public License (LGPL), которая разрешает использование в проприетарных приложениях.

Для отображения графиков с помощью JFreeChart необходимо создать набор данных. Набор данных хранится в объекте класса XYDataset.

Для построения различных графиков JFreeChart (гистограммы, круговые диаграммы, линейные графики) используется ChartFactory, содержащий большое количество статических перегруженных методов.

Для отображения графических объектов в интерфейсе Swing-приложений создаем объект класса ChartPanel, который унаследован от класса JPanel библиотеки swing, что позволяет без труда отобразить график на пользовательском интерфейсе [\[10\]](#).

Студент гр. Б8418

Ильинов
подпись

«12» 06 2018 г.

Научный руководитель к.ф.-м.н., доцент
(должность, учесное звание)

Пустовалов
(подпись)

Е.В. Пустовалов
(и.о.ф.)

«13» 06 2018 г.

Защищена в ГАК с оценкой хорошо

Секретарь ГАК

М
подпись

Тюникова Л.Н.

И.О.Фамилия

«28» июня 2018 г.

«Допустить к защите»

Заведующий кафедрой к.ф.-м.н., доцент

Пустовалов
(подпись)

Е.В. Пустовалов
(и. о.ф.)

«13» 06 2018 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»

ШКОЛА ЕСТЕСТВЕННЫХ НАУК

Кафедра компьютерных систем

ОТЗЫВ РУКОВОДИТЕЛЯ

на выпускную квалификационную работу студента Шевцова Романа Александровича
по основной образовательной программе подготовки бакалавров по направлению
09.03.02 - Информационные системы и технологии группа Б8418

Руководитель ВКР к.ф.м.н., доцент Е.В.Пустовалов

на тему РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА ДЛЯ СИСТЕМЫ
АВТОМАТИЧЕСКИХ МЕХАНИЧЕСКИХ ИСПЫТАНИЙ

Дата защиты ВКР «21» июня 2018г.

Развитие исследовательского комплекса ДВФУ предлагает исследователям новые инструменты, но стандартная автоматизация не всегда отвечает запросам ученых. Автоматизация исследовательского оборудования является актуальной задачей.

Целью работы является разработка программного комплекса системы автоматических механических испытаний. Поставлены следующие задачи: изучение инструментов разработки программного комплекса; найти существующие и создать собственные архитектурные решения для разрабатываемой системы; реализовать программный комплекс.

В результате были разработаны серверное и клиентское приложения, соответствующие техническому заданию. Было проведено эксплуатационное тестирование, для которого были написаны модули, имитирующие работу устройств.

Выпускная квалификационная работа полностью соответствует заданию.

Результаты будут использованы в лабораторном комплексе кафедры компьютерных систем. Дипломник продемонстрировал высокую степень самостоятельности, умение анализировать, обобщать, делать выводы. В ходе выполнения работы Шевцова Р.А. проявил ответственность и работоспособность.

К недостаткам можно отнести ошибки в оформлении, многочисленные орфографические ошибки, а также отсутствие тестов на производительность системы.

Несмотря на указанные недостатки Шевцова Р.А. заслуживает присвоения соответствующей квалификации, а дипломная работа оценки «отлично».

Руководитель ВКР к.ф.м.н., доцент

Е.В.Пустовалов

«20» июня 2018г.