**СОДЕРЖАНИЕ**

Анализ возможностей оптимизации архитектуры сервиса автоматизированного нормоконтроля

# ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

В настоящем отчете применяют следующие термины с соответствующими определениями.

Адаптивный дизайн — это конфигурация, в которой сервер отправляет одинаковый HTML-код на любые устройства, а размеры элементов страницы корректируются с помощью CSS

Веб-приложение (англ. web application) — клиент-серверное приложение, в котором клиент взаимодействует с веб-сервером при помощи браузера

Гибридное приложение (англ. hybrid application) — это скорее подход к программированию, чем какой-то особенный вид приложения. Его принцип заключается в том, что всё что можно программируется на стороне сервера, а на стороне клиента остаётся только необходимый минимум

Десктоп (англ. desktop computer) — стационарный персональный компьютер

Лид — это целевое действие пользователя на сайте или обращение в компанию

Нативное приложение (англ. native application) — это прикладные программы, которые были разработаны для использования на определённой платформе или на определённом устройстве

Паттерн (англ. pattern) — образец, шаблон, повторяемая архитектурная конструкция, представляющая собой решение проблемы проектирования

E-commerce (электронная коммерция) — это сфера экономики, которая включает в себя все финансовые и торговые транзакции, осуществляемые при помощи компьютерных сетей, и бизнес-процессы, связанные с проведением таких транзакций

User Flow (пользовательские сценарии) — переход пользователей от одного сценария взаимодействия с интерфейсом к другому, пользовательский маршрут

Usability (юзабилити) — способность продукта быть эргономичным, понимаемым, изучаемым, используемым и привлекательным для пользователя в заданных условиях

# ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

БД — база данных

ОС — операционная система

РФ — Российская Федерация

API (англ. application programming interface) — программный интерфейс приложения, набор готовых классов, функций, процедур, структур и констант.

CMS (англ. content management system) — это система управления контентом, набор скриптов для создания, редактирования и управления контентом сайта. На профессиональном жаргоне CMS также называют «движок». Примерами CMS являются WordPress, Joomla, OpenCart.

MVC (англ. model-view-controller) — схема разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: модель, представление и контроллер

PSI (англ. page speed insights) — API (PSI) позволяет получать отчеты о скорости загрузки страниц на мобильных устройствах и компьютерах, а также советы, как эту скорость увеличить.

PWA (англ. progressive web application) — технология в web-разработке, которая визуально и функционально трансформирует сайт в приложение (мобильное приложение в браузере).

UI (англ user interface design) — это то, как выглядит интерфейс и то, какие физические характеристики приобретает. Определяет, какого цвета будет ваше «изделие», удобно ли будет человеку попадать в кнопки, читабельным ли будет текст и тому подобное.

UX (англ. user experience design) — это то, какой опыт/впечатление получает пользователь от работы с интерфейсом. Удается ли ему достичь цели и на сколько просто или сложно это сделать.

# ВВЕДЕНИЕ

Тема индивидуального задания: Определение направления магистерского исследования и разработка его программы

1. Определить тематику исследования. Сформулировать тему исследования (предварительную) в соответствии с тематикой.
2. 1.Изучить не менее 20 источников информации по теме исследования, в том числе не менее 3 на иностранном языке. 2. По результатам исследования источников информации уточнить тему магистерской диссертации. 3. Аргументированно доказать актуальность темы исследования.
3. Сформулировать методологические элементы исследования. 4. Сформулировать методологические элементы исследования, в частности определить: – цель и задачи магистерской диссертации; – актуальность темы исследования; – степень разработанности темы исследования; – новизна и значимость предполагаемых результатов работы; – методы исследования, которые предполагается использовать в ходе исследования; – как будет доказываться степень достоверности полученных результатов.

– цель и задачи магистерской диссертации; – актуальность темы исследования; – степень разработанности темы исследования; – новизна и значимость предполагаемых результатов работы; – методы исследования, которые предполагается использовать в ходе исследования; – как будет доказываться степень достоверности полученных результатов.

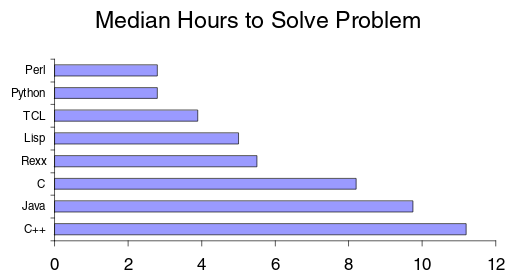
Разработать программу исследования

# Цифровая трансформация бизнеса

Производительность – это не только функция скорости исполнения языка, реализации программы, а производительность сторонних библиотек обычно является фактором номер один в уравнении.

Решение использовать интерпретируемый язык программирования в высокопроизводительных приложениях может быть парадоксальным, но мы столкнулись с тем, что CPU редко, когда является сдерживающим фактором; выразительность языка означает, что большинство программ невелики и большую часть времени тратят на ввод-вывод, а не на собственный код. Более того, гибкость интерпретируемой реализации была полезной, как в простоте экспериментов на лингвистическом уровне, так и в предоставлении нам возможности исследовать способы распределения вычислений на многих машинах. [22]

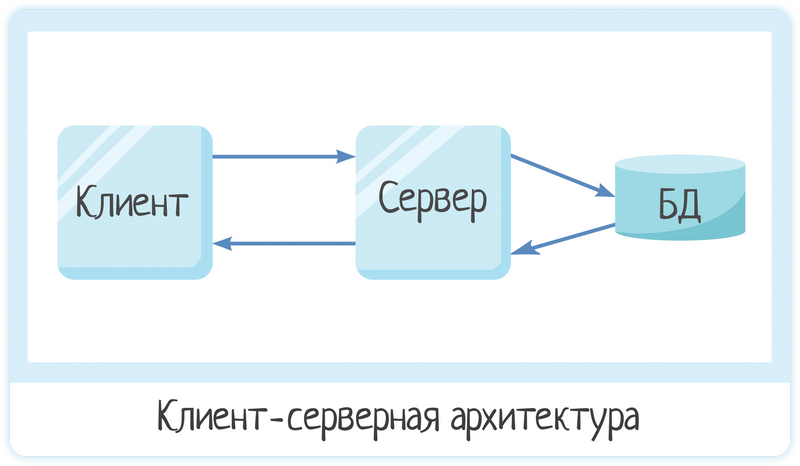
Что насчёт таких аргументов: «Всё это прекрасно, но что, если CPU становится узким местом и это начинает сказываться на производительности?» или «Язык x менее требователен к железу, нежели y»? Они тоже имеют место быть, однако вы можете масштабировать приложение горизонтально бесконечно. Однако это палка о двух концах…

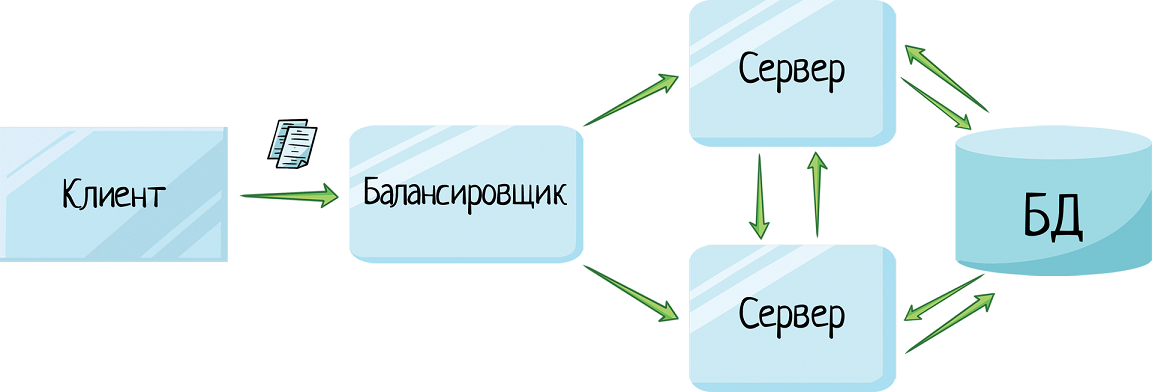
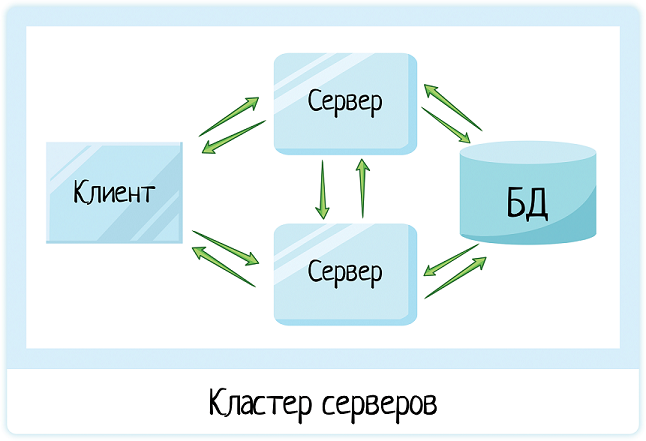


В представленном выше исследовании Python более чем в 2 раза продуктивнее Java. Многие другие также языки программирования дают похожий результат. Rosetta Code провёл довольно обширное исследование различий изучения языков программирования. В статье они сравнивают python с другими интерпретируемыми языками и заявляют:

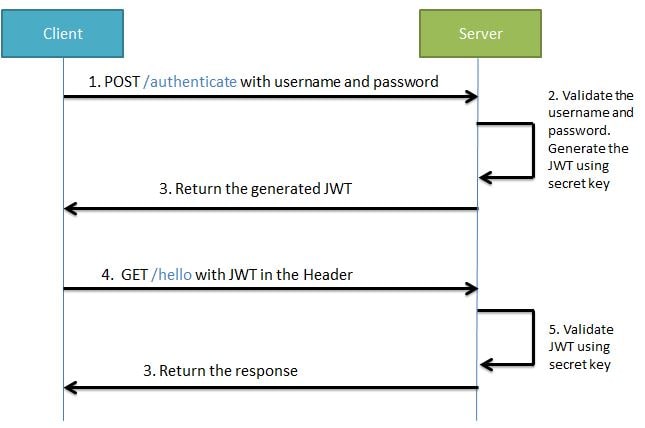
Python, в целом, наиболее краток, даже в сравнении с функциональными языками (в среднем в 1,2-1,6 раза короче).

По-видимому, в реализации на Python будет как правило меньше строк кода, чем на каком-либо другом языке. Это может показаться ужасной метрикой, однако несколько исследований, в том числе и упомянутых выше, показывают, что время, затраченное на каждую строку кода, примерно одинаково на каждом языке. Таким образом, чем меньше строк кода, тем больше продуктивность.





<https://habr.com/ru/post/495698/>





<https://dzone.com/articles/spring-boot-security-json-web-tokenjwt-hello-world>

# ЗАКЛЮЧЕНИЕ

В результате дипломной работы доработанный веб-ресурс был развёрнут на хостинге timeweb и доступен по адресу:

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Мобильные приложения: Native vs Hybrid. Что выбрать? [Электронный ресурс]. Режим доступа: <https://fountain.company/mobilnye-prilozheniya-native-vs-hybrid-chto-vybrat/> (дата обращения: 20.04.2020).
2. Нарчук, О. А. Разработка клиент-серверного приложения средствами Java / О. А. Нарчук // Компьютерное проектирование и технология производства электронных систем: сборник материалов 50-й научной конференции аспирантов, магистрантов и студентов (Минск, 24-28 марта 2014 г.). – Минск: БГУИР, 2014.
3. Мартин Р. Чистый код. Создание, анализ и рефакторинг — СПб.: Питер, 2019. — 464 с.: ил.
4. Лафоре Р. Структуры данных и алгоритмы в Java. Классика Computers Science. 2-е изд. — СПб.: Питер, 2013. — 704 с.: ил.
5. Анкудинов Г.И., Стрижаченко А.И. Сети ЭВМ и телекоммуникации. Архитектура и протоколы: Учеб. пособие. - СПб.: СЗТУ, 2001, - 92 c.
6. Ананьев П.И., Кайгородова М.А. Основы баз данных.: Учебное пособие/ Алт. госуд. технич. ун-т им. И.И. Ползунова. - Барнаул: 2010. - 189. - ил.
7. Бхаргава А., Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих. - СПб.: Питер, 2017. - 288 с.: ил.
8. Раджпут Динеш, Spring. Все паттерны проектирования. — СПб.: Питер, 2019. — 320 с.: ил.
9. Блох, Джошуа, Java: эффективное программирование, 3-е изд.: Пер. с англ. — СПб.: ООО “Диалектика”, 2019. — 464 с.: ил.
10. Хемраджани А., Гибкая разработка приложений на Java с помощью Spring, Hibernate и Eclipse.: Пер. с англ. – М.: ООО «И. Д. Вильямс», 2008. -352с.: ил.
11. Бройдо В. Л., Ильина О. П., Вычислительные системы, сети и телекоммуникации - СПб.: Питер, 2011 – 560 с.: ил.
12. Комагоров В.П., Архитектура сетей и систем телекоммуникаций: учебное пособие / В.П. Комагоров; Томский политехнический университет. – Томск: Изд-во Томского политехнического университета, 2011. – 154 с.
13. Libor Dostálek, Alena Kabelová, Understanding TCP/IP A clear and comprehensive guide to TCP/IP protocols, 2006. – 435с.: ил.
14. Calvert K. L., Donahoo M. J., TCP/IP Sockets in Java: Practical Guide for Programmers, Second Edition, 2008.
15. Williams N. S., Professional Java for Web Applications, 2014.
16. Macero M., Learn Microservices with Spring Boot, 2017.
17. Лонг Д., Бастани К., Java в облаке. Spring Boot, Spring Cloud, Cloud Foundry. – СПб.: Питер, 2019 – 624с.: ил.
18. Эккель Б., Философия Java. Библиотека программиста. 4-е изд. – СПб.: Питер, 2009 640с.: ил.
19. Saternos С., Client-Server Web Apps with JavaScript and Java
20. M.-Leander Reimer, Building RESTful Web Services with Java EE 8
21. K. Siva Prasad Reddy, Beginning Spring Boot 2: Applications and Microservices with the Spring Framework
22. Rob Pike, Sean Dorward, Robert Griesemer, Sean Quinlan, Interpreting the Data: Parallel Analysis with Sawzall- Google, Inc.
23. Sebastian Nanz, Carlo A. Furia, A Comparative Study of Programming Languages in Rosetta Code

# ПРИЛОЖЕНИЕ А

**Заголовок**