

бинарный поиск. А табличный метод наиболее эффективен при любом диапазоне выборки.

Практическое применение данных алгоритмов я осуществил на примере выборки студентов по среднему баллу. Мною было решено применить только линейный и табличный методы как более интересные для анализа результатов, исключив бинарный по причине его крайней неэффективности при решении этой задачи. Как и предполагалось, поиск по табличному алгоритму оказался более быстрым и эффективным.

#### ЛИТЕРАТУРА

1. Кнут Д. Э. Искусство программирования. Том 3. Сортировка и поиск / под ред. В. Т. Тертышного (гл. 5) и И. В. Красикова (гл. 6). — 2-е изд. — Москва: Вильямс, 2007. — Т. 3. — 832 с.

2. Макконнелл С. Совершенный код. Мастер-класс / Пер. с англ. — издательство «Русская редакция», 2010. — 896 стр.

3. CodeLessons Бинарный поиск по массиву C++ [Электронный ресурс]. — Режим доступа: <https://codelessons.ru/cplusplus/algoritmy/binarnyj-poisk-po-massivu-c.html>. — Дата доступа: 15.03.2021

УДК 004.415.2

Студ. Е.А. Гончар

Науч. рук. зав. кафедрой Н.В. Пацей  
(кафедра программной инженерии, БГТУ)

### СИСТЕМА ОТСЛЕЖИВАНИЯ ОШИБОК И УПРАВЛЕНИЕ ПРОЕКТОМ

Система отслеживания ошибок (англ. *bug tracking system*) – прикладная программа, разработанная с целью помочь разработчикам программного обеспечения (программистам, тестировщикам и др.) учитывать и контролировать ошибки (баги), найденные в программах, а также следить за процессом устранения этих ошибок.

Целью работы была разработка веб-приложения «*Issue Tracking System*». Приложение содержит две части: серверную, разработанную на языке *Java* и фреймворка *Spring*, и клиентскую, в которой использовался *Java Script* с фреймворком *Angular*.

Функционально приложение содержит две роли пользователь и администратор. Пользователи без прав администратора могут создавать, редактировать, удалять и фильтровать задачи и ошибки, а также управлять профилем. Администратор может выполнять все вышеперечисленные функции, а также создавать новых пользователей, редактировать информацию о них и удалять.

Структура пакетов приложения представлена на рисунке 1.

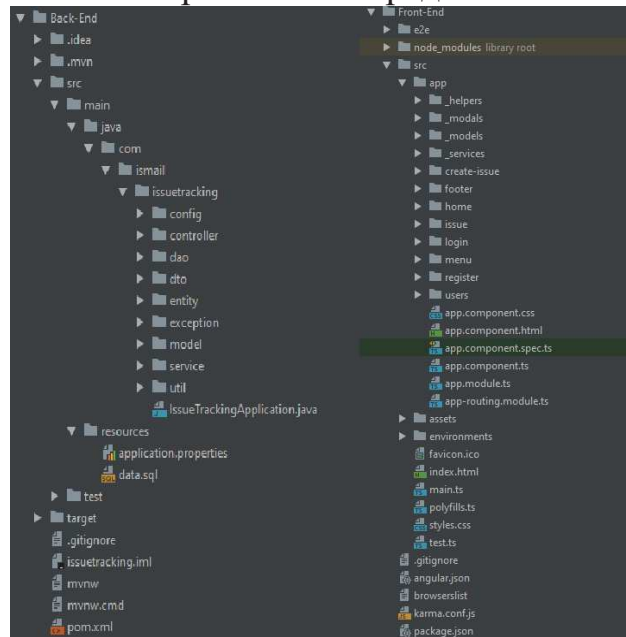


Рисунок 1 – Структура пакетов приложения

Для создания веб-приложения и в дальнейшем получения и редактирования информации об ошибках и заданиях были разработаны следующие сущности:

- *Issues* – в данной сущности находится вся основная информация об ошибках такая как дата создания, имя создателя, имя того на кого привязана данное задание и т.д.;

- *Position* – который определяет статус пользователя в общей иерархии системы предприятия;

- *Role* – определяет роль пользователя, в зависимости от которой будут выдаваться соответствующие права;

- *Status* – в данной сущности хранится информация о том, на какой стадии находится процесс исправления ошибки или выполнения задания;

- *Type* – тип хранит указание на то, что мы исправляем или выполняем;

- *User* – хранит информацию о юзере и всех элементах связанных с ним.

В приложении разработаны следующие контроллеры:

- *Authentication Controller* – это контроллер обеспечивающий безопасность приложения, а так же все что связано с выдачей токена для дальнейшего использования приложения;

- *Issues Controller* – контроллер цель которого связать данные из базы данных про все ошибки и задания с частью приложения отвечающей за визуализацию;

- User Controller – контроллер использующийся для взаимодействия с сущностью User в которой хранится информация про всех пользователей, а также участвующий в месте с Authentication Controller в процессе авторизации.

Разработанное приложение может быть использовано для небольших компаний при ведении и управлении программных проектов.

УДК 316.667

Магистрант П.И. Карпович  
Науч. рук. зав. кафедрой Н.В. Пацей  
(кафедра программной инженерии, БГТУ)

## **ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ МОДЕЛЬ УПРАВЛЕНИЯ ИМЕНОВАННЫМИ ДИНАМИЧЕСКИМИ ОБЪЕКТАМИ КОНТЕНТ-ОРИЕНТИРОВАННОЙ СЕТИ (ICN)**

В настоящее время на передачу мультимедийного контента, такого как: фильмы, музыкальные клипы, теле- и радиoprogramмы, приходится около 70% интернет-трафика. Интернет сеть не обеспечивает эффективное распространение такого рода контента, поэтому необходимо разрабатывать новые решения [1].

Существует направление исследований, связанное с улучшением распространения информации через контент-ориентированные сети - ICN (*Information-Content Net working*) [1-2].

Среди актуальных направлений исследований: сетевые механизмы и алгоритмы ICN; публикация контента; маршрутизация и передача; алгоритмы выбора источника; управление кэшированием и др. В связи с этим, особую актуальность имеют исследования эффективности прототипов ICN в условиях, близких к реальным, а также измерения скорости передачи данных, заполнения и политики использования кэша, вероятности отказов.

В настоящей работе выполнено проектирование и разработка узлов контент-ориентированной сети на языке программирования Go.

Система содержит три основных типа узлов: *content-provider*, *content-manager*, *client* и ряд служебных. В качестве объектов распространения используются файлы.

Каждый *content-provider* содержит список файлов, локальный кэш и осуществляет коммуникации с *client* и другими *content-provider*. *Content-manager* основан на слушателях и поддерживает четыре типа команд: *connect*, *disconnect*, *providers* и *query*.

За счет того, что каждый из описанных узлов разворачивается в отдельном *docker*-контейнере, модель масштабируется в соответствии