

Направления для НИР (магистры)

[Блеес] Система автоматической проверки наиболее частых формальных ошибок в научных текстах

Цель: разработать настраиваемый статический анализатор для формальных текстов - научно-технических статей, отчетов и дипломов. Анализ происходит по набору наиболее частых ошибок, которые при этом являются машинно-проверяемыми, например:

1. личные предложения, формы глаголов и местоимения;
2. отсутствие ссылок или битые ссылки на элементы списка литературы/рисунки/таблицы;
3. отсутствие упоминания ключевых слов в тексте статьи;
4. повторы слов в пределах двух предложений;
5. «телеграфность» - повторение начальных слов абзацев («Было принято решение»);
6. стоп-слова:
 1. жаргонизмы (использовать список): скачать, пост, либа, тул;
 2. там, тут, здесь.

Задачи:

1. изучение как ближних (например spellcheck), так и дальних аналогов (lint'ы и статические анализаторы для языков программирования);
2. формулировка сценариев использования и создание макета UI;
3. разработка модели данных для правил;
4. парсинг docx/ppt/pdf;
5. полнотекстовый поиск;
6. хранение и эффективная интерпретация правил;
7. создание веб-интерфейса для инструмента;
8. аннотирование проверяемого документа (например добавление комментариев прямо в doc(x));

Требования:

- начальный опыт написания формальных текстов и примерное понимание, почему нужно выполнять большую часть правил, описанных в цели работы;
- опыт обработки машино-читаемых документов (HTML, XML, JSON, CSV);
- начальные знания Python;
- начальные знания веб-технологий.

Результат (НИР): исследование существующих алгоритмов анализа структуры текстов, сценарии командной строки для автоматизации проверки.

[Чирухин] Автоматизация создания плагинов Vagrant

Цель: автоматизировать создание и публикацию плагинов Vagrant.

Задачи:

- изучение процедуры разработки и публикации плагинов на примере типовых задач:

- создание своей команды vagrant,
- вызов команд shell,
- получение доступа к структурам данных vagrant;
- автоматизация создания плагинов с помощью сценариев командной строки, шаблонов;
- создание плагинов-примеров;
- создание руководства по разработке и публикации;
- публикация руководства на stepik.org в формате мини-курса.

Требования:

- Ruby.
- Опыт использования Vagrant.

Результат (НИР):

- инструменты автоматизации разработки плагина;
- плагины-примеры;
- руководство по процессу.

Темы ВКР бакалавров

[Андросов] Автоматическая система проверки задач для MOOC "Мобильная разработка для Android на Kotlin"

Цель: создание набора автоматически проверяемых лабораторных по мобильной разработке на Kotlin, интегрированных в stepik.org.

Задачи:

- обзор онлайн-курсов для изучения Java+Android, Kotlin, Kotlin+Android и сравнение предлагаемых задач;
- изучение интерфейса командной строки для сборки, запуска и тестирования мобильных приложений в эмуляторе Android, автоматизация типичных сценариев использования;
- подготовка образа виртуальной машины для задач курса, содержащего необходимые инструменты и среду;
- разработка неинтерактивных задач:
 - описание условий задач,
 - создание эталонных решений (правильных и содержащих различные ошибки),
 - создание скриптов проверки задач (на Bash и Espresso)
- интеграция наработок на Stepik.org;
- исследование производительности полученного решения.

Требования:

- опыт Android разработки и интерес к данной предметной области;
- начальное представление о работе виртуальных машин;
- начальные знания Bash.

Результат: набор автоматически проверяемых задач с описаниями, подключенных к курсу на Stepik.org.

[Хафизов] Автоматизация проверки работ в очных курсах по мобильной разработке

Идея: на кафедре проходит много мероприятий, на которых происходит одновременная разработка и мобильных приложений. Для таких мероприятий необходим инструмент, позволяющий проверить в динамике:

1. какие из приложений собираются,
2. какие из приложений соответствуют требованиям,
3. объем арк файлов,
4. успешность установки на устройство,
5.

Цель: разработать инструмент автоматизации сборки и тестирования мобильных приложений для набора репозитория.

Задачи:

1. запуск сборки и тестирования в jenkins
2. генерация итоговых отчетов с результатами сборки и/или тестирования по набору репозитория
3. создание арк и их публикация

Требования:

1. начальные знания Android-разработки,
2. начальные знания Bash,
3. минимальный опыт в Jenkins,

Результат: инструмент, позволяющий автоматически оценивать качество и репозитория с исходниками мобильных приложений путем их сборки/тестирования/установки; инструкции по развертыванию и настройке инструмента.

[Козловских] Инструмент разработки, моделирования и модерирования outdoor-квестов

Цель: разработать веб-сервис создания сценариев для outdoor-квестов.

Задачи:

- Исследовать примерные аналоги - приложения для составления маршрутов на карте, визуализаторы gps-треков и т.д.
- Разработать веб-интерфейс, который позволит
 - осуществлять CRUD для квестов;
 - настраивать генератор квестов и создавать квесты с его помощью;
 - визуализировать квесты на карте,
 - проводить анализ квестов и их сравнение:
 - примерная длительность прохождения,
 - протяженность,
 - потенциальная сложность;

- моделировать прохождение квеста по следующим характеристикам (их может быть больше - ниже примеры):
 - оценивать потенциальный граф перемещений (насколько сильно человек может плутать в поисках нужных точек),
 - распределение времени прохождения квеста.
- предоставлять интерфейс модерирования квестов;
 - предоставлять интерфейс для ручного создания/редактирования квестов;
 - выбирать источники данных для подбора точек.
- Подключиться к интерфейсам генератора, передавать ему правила генерации и получать созданные квесты.

Требования:

- Python + TurboGears.
- Bootstrap
- Leaflet.js

Результат: веб-приложение, позволяющее не знакомому с технической частью пользователю быстро создавать свои квесты либо модерировать и править квесты, созданные генератором.

From:

<https://se.moevm.info/> - **МОЭВМ Вики** [se.moevm.info]

Permanent link:

https://se.moevm.info/doku.php/staff:courses:theses_zmm_2017

Last update:

