SRS.PI.01.01

Учебное приложение «Integrator» («Интегратор»)

Назначение

Учебное приложение «Integrator» (в дальнейшем приложение) предназначено для следующих целей:

- ✓ Изучение программирования на платформе Java с использованием фреймворка Spring;
- ✓ Изучение основных методов интеграции данных в рамках науки о данных;
- ✓ Освоение основ тестирования программного обеспечения.

Общее описание приложения

Приложение предназначено для сбора данных в интернете по запросу пользователя приложения. Запрос оформляется в виде задачи на поиск информации в формате текстового файла.

Общий порядок работы приложения:

1. При старте приложения пользователь указывает файл с описанием задачи на поиск информации, которую нужно найти в интернете;
2. Приложение считывает файл с определением задачи в виде набора параметров;
3. Приложение преобразует набор параметров в модель данных;
4. На основе анализа модели данных приложение составляет поисковый запрос.
5. Составленный поисковый запрос приложение отправляет в поисковую систему Яндекса;
6. Полученный ответ поисковой системы оценивается с целью выбора заданного количества наиболее приоритетных ссылок;
7. Каждая из отобранных ссылок используется приложением для получения веб-страницы, адрес которой определяется этой ссылкой;
8. Каждая полученная веб-страница обрабатывается как отдельный источник с целью извлечения данных в соответствии с заданной моделью;
9. Извлечённая информация заносятся в интегрированный набор данных.
10. После обработки всех отобранных источников приложение формирует ответ в виде текстового файла с информацией, которая соответствует заданной пользователем модели данных.

Приложение разрабатывается в двух версиях, основные различия между которыми заключены в реализации пользовательского интерфейса:

Версия 1.0 имеет консольный интерфейс (интерфейс командной строки);

Версия 2.0 имеет веб-интерфейс.

Используемые термины, понятия и обозначения

Модель данных - описание объекта и его атрибутов в виде структурированного текстового файла. Модель данных определяет задачу на поиск информации.

Поисковая система - внешний веб-сервис, который используется приложением для поиска источников информации в интернете.

Требования

Функциональные требования

1. Приложение считывает описание модели данных из текстового конфигурационного файла.
2. Конфигурационный файл приложения с описанием модели данных определяется одним из следующих способов:
   2.1. По умолчанию используется имя файла properties.cfg, который должен располагаться в корневом каталоге приложения;
   2.2. В версии 1.0 может задаваться первым после имени стартового класса аргументом командной строки, например,

```
java Integrator result.txt;
```

   2.3. В версии 2.0 задаётся пользователем в отдельном поле окна приложения.
3. Приложение автоматически преобразовывает текстовое описание модели данных во внутреннее представление.
4. Приложение автоматически формирует запрос к поисковой системе на основе заданной модели данных.
5. Приложение автоматически выбирает из ответа поисковой системы ссылки на ресурсы.
6. Приложение автоматически оценивает приоритет ресурса из ответа поисковой системы.
7. Приложение автоматически получает ресурсы (документы) по ссылкам из ответа поисковой системы.
8. Приложение выбирает из полученных ресурсов информацию для заполнения заданной модели данных.
9. Результаты работы приложения выводятся в отдельный текстовый файл.

10. Имя и расположение файла результата работы приложения задаётся пользователем:

10.1. В версии 1.0 в конфигурационном файле параметром file.output или аргументом командной строки;

10.2. В версии 2.0 - в отдельном поле окна приложения.

Нефункциональные требования

Требования к данным

11. Модель данных описывается в отдельном текстовом файле.
12. Файл описания задачи на поиск данных содержит:
    12.1. Заголовок;
    12.2. Набор элементов с описанием задачи поиска;
    12.3. Набор элементов, определяющих искомую информацию.
13. Каждый элемент задачи на поиск данных определяется в формате:
    13.1. ключ = значение
14. Элементы задачи поиска должны иметь установленное не пустое значение.
15. Элементы, определяющие искомую информацию, могут иметь пустое значение.

Общее описание архитектуры приложения

Общая структура и назначение отдельных компонентов приложения:

- ✓ Класс Integrator является стартовым классом приложения, который обеспечивает его запуск, первоначальную настройку и переход в режим обработки запросов пользователей.
- ✓ Интерфейс UI определяет набор методов для взаимодействия с пользовательским интерфейсом приложения (в первой версии приложения не представлен).
- ✓ Интерфейс Task определяет методы работы с параметрами задачи на поиск информации.
- ✓ Интерфейс Preprocessor определяет методы работы с поисковым сервисом и формирование списка источников для последующего их анализа.
- ✓ Интерфейс Analyzer определяет методы анализа и извлечения данных из заданного источника.
- ✓ Интерфейс Reporter определяет методы формирования итогового отчёта по обработке задачи на поиск информации.
- ✓ Интерфейс Sorce определяет методы работы с отдельным источником данных.

- ✓ Функциональный интерфейс Converter определяет методы преобразования наборов данных из одного вида/формата в другой.
- ✓
- ✓ Интерфейс Data представляет набор данных.
- ✓ Интерфейс Mode содержит набор статических констант, используемых для определения различных стадий обработки или очистки наборов данных.
- ✓ Интерфейс Model представляет методы для описания сложных моделей данных(в первой версии приложения не представлен).
- ✓ Класс IntegratorException представляет базовое исключение приложения.
- ✓ Функциональный интерфейс Validator определяет методы оценки качества наборов данных (в первой версии приложения не представлен). Данные методы могут использоваться на любой стадии обработки поставленной задачи. Например, методы могут применяться для предварительной оценки наборов данных, извлечённых из источников данных, или для оценки итогового набора данных.