Домашнее задание 9. Web Crawler

1. Напишите потокобезопасный класс `WebCrawler`, который будет рекурсивно обходить сайты.
    1. Класс `WebCrawler` должен иметь конструктор

```
public WebCrawler(Downloader downloader, int downloaders, int extractors, int perHost)
```

        - `downloader` позволяет скачивать страницы и извлекать из них ссылки;
        - `downloaders` — максимальное число одновременно загружаемых страниц;
        - `extractors` — максимальное число страниц, из которых извлекаются ссылки;
        - `perHost` — максимальное число страниц, одновременно загружаемых с одного хоста. Для опредения хоста следует использовать метод `getHost` класса `URLUtils` из тестов.
    2. Класс `WebCrawler` должен реализовывать интерфейс `Crawler`

```
public interface Crawler extends AutoCloseable {
    Result download(String url, int depth);

    void close();
}
```

        - Метод `download` должен рекурсивно обходить страницы, начиная с указанного URL на указанную глубину и возвращать список загруженных страниц и файлов. Например, если глубина равна 1, то должна быть загружена только указанная страница. Если глубина равна 2, то указанная страница и те страницы и файлы, на которые она ссылается и так далее. Этот метод может вызываться параллельно в нескольких потоках.
        - Загрузка и обработка страниц (извлечение ссылок) должна выполняться максимально параллельно, с учетом ограничений на число одновременно загружаемых страниц (в том числе с одного хоста) и страниц, с которых загружаются ссылки.
        - Для распараллеливания разрешается создать до `downloaders + extractors` вспомогательных потоков.
        - Загружать и/или извлекать ссылки из одной и той же страницы в рамках одного обхода (`download`) запрещается.
        - Метод `close` должен завершать все вспомогательные потоки.
    3. Для загрузки страниц должен применяться `Downloader`, передаваемый первым аргументом конструктора.

```
public interface Downloader {
    public Document download(final String url) throws IOException;
}
```

- Метод `download` загружает документ по его адресу (URL).
- Документ позволяет получить ссылки по загруженной странице:

```
public interface Document {
    List<String> extractLinks() throws IOException;
}
```

Ссылки, возвращаемые документом являются абсолютными и имеют схему `http` или `https`.

4. Должен быть реализован метод `main`, позволяющий запустить обход из командной строки
   - Командная строка

```
WebCrawler url [depth [downloads [extractors [perHost]]]]
```

   - Для загрузки страниц требуется использовать реализацию `CachingDownloader` из тестов.

2. Версии задания
   1. *Простая* — можно не учитывать ограничения на число одновременных закачек с одного хоста `(perHost >= downloaders)`.
   2. *Полная* — требуется учитывать все ограничения.
   3. *Бонусная* — сделать параллельный обод в ширину.