

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده ریاضی و علوم کامپیوتر

استاد درس: دکتر مهدی قطعی

استاد کارگاه: بهنام یوسفی مهر

طراح پروژه و نگارنده: پارسا حسینی‌پور رفسنجانی

بهار ۱۴۰۳

راهنمای پیاده‌سازی RSS Reader

درس برنامه‌سازی پیشرفته و کارگاه

همه‌ی ما از وبسایت‌ها و وبلاگ‌های متفاوتی بازدید و از اطلاعات متفاوتی که در این منابع ارائه می‌شود استفاده می‌کنیم. رویه‌ی معمول به این صورت است که در موتورهای جست‌وجو مثل گوگل مطالب مورد نظر خود را جست‌وجو و از آن طریق از مطالب مرتبطی که افراد گوناگون نوشته‌اند بازدید می‌کنیم. با این حال شاید در این بین وبسایت‌های جالبی را یافته و بخواهیم مطالبشان را دنبال کنیم. برای این کار می‌توانیم صفحات مجازی نویسنده‌های این وبسایت‌ها را دنبال کنیم، اما شاید همه‌ی مطالب خود وبسایتشان در صفحه‌ی مجازی‌شان بازنشر نشود، و یا شاید اصلاً نویسندگان صفحه‌ی مجازی نداشته باشند! ممکن است بگویید می‌توانیم عضو خبرنامه‌شان شویم و ایمیل‌هایشان را دریافت کنیم، ولی حتی اگر وبسایتی خبرنامه هم داشته باشد، در صورت عضویت ممکن است تنها در چند روز در میان حجم عظیمی از ایمیل‌ها آن ایمیل‌های وبسایت را یا خودمان نادیده بگیریم و یا سرویس ایمیل آنها را به عنوان اسپم شناسایی کند. در این حالت باید چه کار کنیم تا مطالبشان را از دست ندهیم؟ در ادامه با مفهوم RSS آشنا می‌شویم و با استفاده از این مفهوم برنامه‌ای می‌نویسیم تا بتوانیم مطالب وبسایت‌های مورد نظرمان را دنبال کنیم.

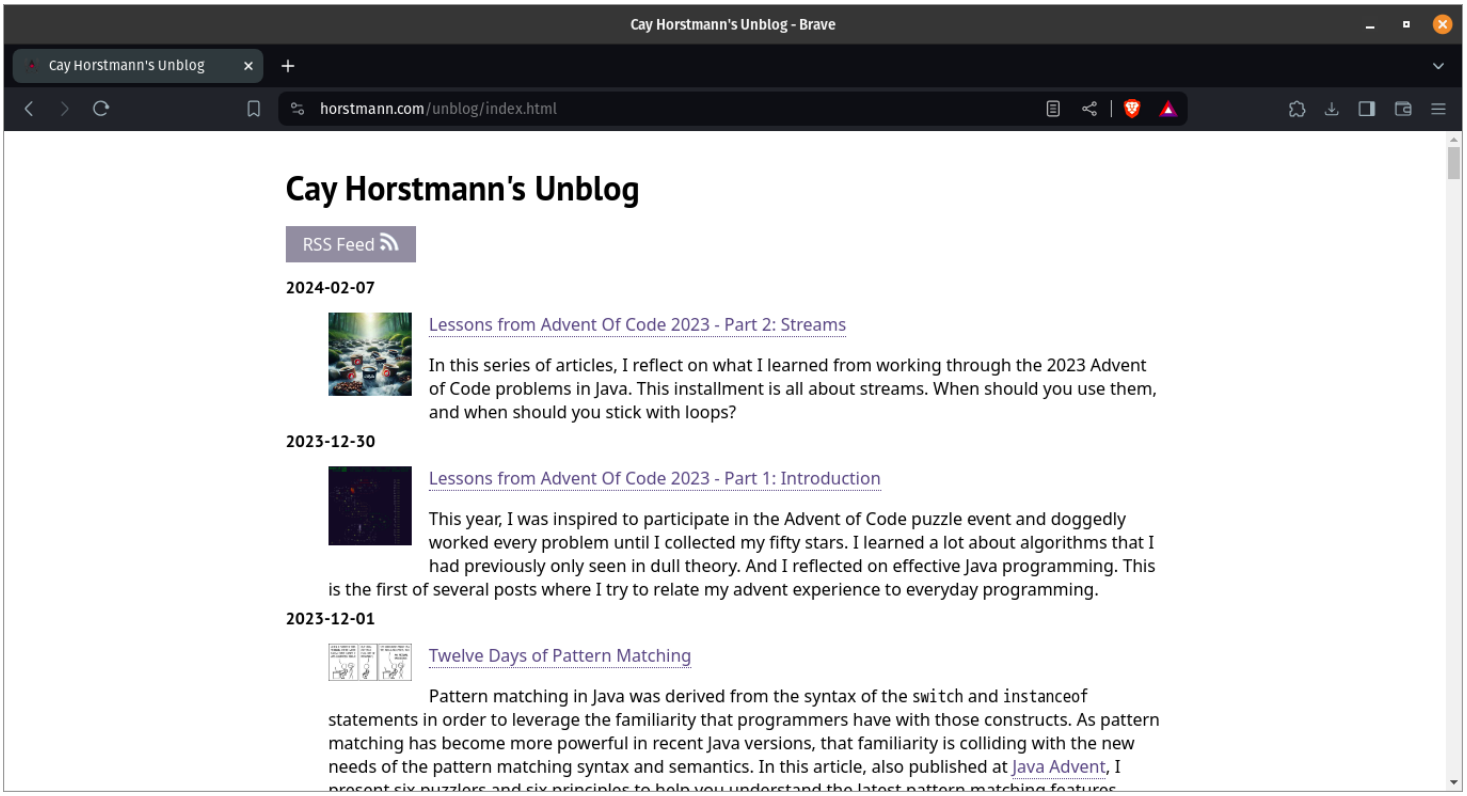
خوراک RSS [1]



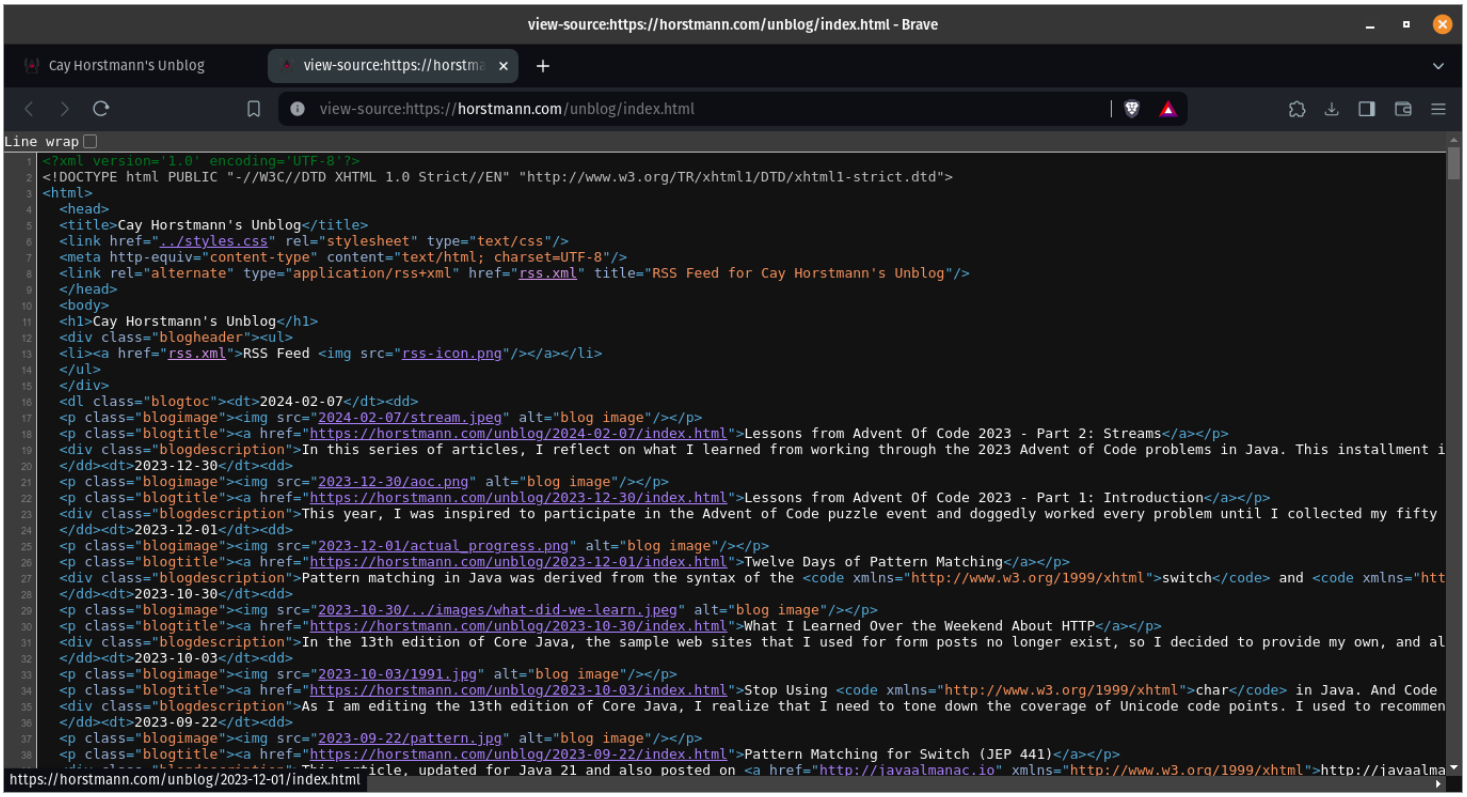
RSS مخفف Really Simple Syndication است که به فایل‌هایی تحت عنوان XML اشاره دارد که به سادگی توسط کامپیوتر خوانده شده و بروزرسانی می‌شوند. برنامه‌هایی وجود دارند که اصطلاحاً RSS Reader یا RSS Feed Reader نامیده می‌شوند و اطلاعات خوراک RSS را از سایت‌های مورد نظر کاربر دریافت کرده و به شکلی که به راحتی قابلیت خواندن داشته باشند به او نمایش می‌دهند. خوراک RSS سرخط و خلاصه‌ی مطالب و بروزرسانی اطلاعیه‌ها و لینک‌های مطالب مربوطه را دربر دارد. توزیع این محتویات بی‌درنگ [2] است و لذا نتایج بالاتر همیشه جدیدترین محتویات منتشر شده خواهند بود.

[1] RSS Feed
[2] real-time

اکثر وبسایت‌ها و پادکست‌ها RSS دارند و شما به راحتی می‌توانید بدون هیچ برنامه‌ای و تنها با دسترسی به اینترنت XML مورد نظرتان را دریافت کنید. در ادامه وبلاگ Cay Horstmann به نشانی https://horstmann.com/unblog را به عنوان مثال در نظر می‌گیریم.



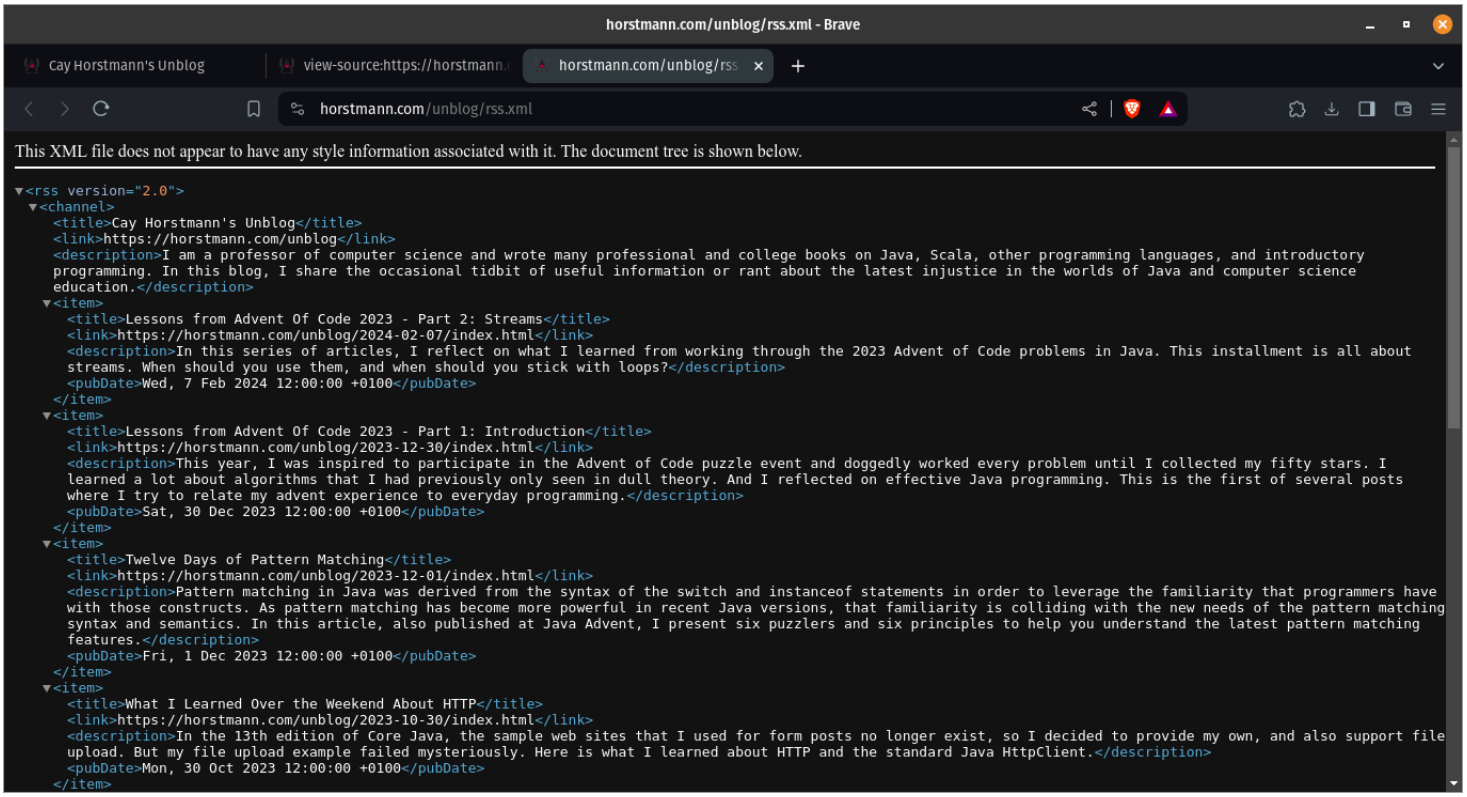
حال روی صفحه کلیک راست کرده و روی View page source یا عنوان مشابهی بسته به نوع مرورگر کلیک می‌کنیم. آنگاه صفحه‌ی index.html وبلاگ مورد نظر را می‌توانیم مشاهده کنیم.



به عنصر link که حاوی اطلاعات مربوط به RSS وبلاگ است توجه کنید. برای اطلاعات بیشتر در مورد این عنصر می‌توانید به اسناد مربوطه‌ی MDN مراجعه نمایید.

```
<link rel="alternate" type="application/rss+xml" href="rss.xml" title="RSS Feed for Cay Horstmann's Unblog"/>
```

اگر روی rss.xml که داخل href است کلیک کنید، صفحه‌ای مشابه صفحه‌ی زیر را خواهید دید.



این همان فایل XML خوراک RSS است. ساختار XML با نظم مشخصی اطلاعات وبلاگ را قابل خواندن می‌کند. در داخل عنصر title نام وبلاگ و در داخل هر عنصر item اطلاعات مربوط به مطالب اخیر وبلاگ آمده است. اطلاعات داخل XML را با صفحه‌ی اصلی خود وبلاگ مقایسه کنید.

برنامه‌ای که طراحی می‌کنید باید کارهایی که در این قسمت انجام دادیم را برای یک وبسایت فرضی به صورت خودکار انجام دهد. برای اتصال به اینترنت از کلاس‌های URL و URI و برای خواندن محتویات html از jsoup استفاده کنید. همچنین نیاز به خواندن اطلاعات سازمان یافته‌ی XML خواهید داشت. برای خواندن و به دست آوردن اطلاعات مورد نیاز از Java DOM Parser استفاده کنید.

ساختار برنامه

برنامه‌ی RSS Reader که می‌خواهیم طراحی کنیم باید در زمان شروع اجرا لیستی از نام‌ها، لیستی از نشانی و لیستی از آدرس RSS وبسایت‌های مورد علاقه‌مان را بارگذاری کرده و در پایان اجرا هرگونه تغییرات را در این لیست‌ها ذخیره کند. به همین دلیل باید از File I/O استفاده کنیم. اطلاعات مربوط به سایت‌ها در فایلی به نام data.txt ذخیره می‌شوند. در هر خط این فایل نام وبسایت، نشانی وبسایت و نشانی خوراک RSS آن قرار دارد که با ; از هم جدا می‌شوند. مثلاً وبلاگی که بررسی کردیم به صورت زیر در این فایل نوشته خواهد شد:

Cay Horstmann’s Unblog;https://horstmann.com/unblog/index.html;https://horstmann.com/unblog/rss.xml

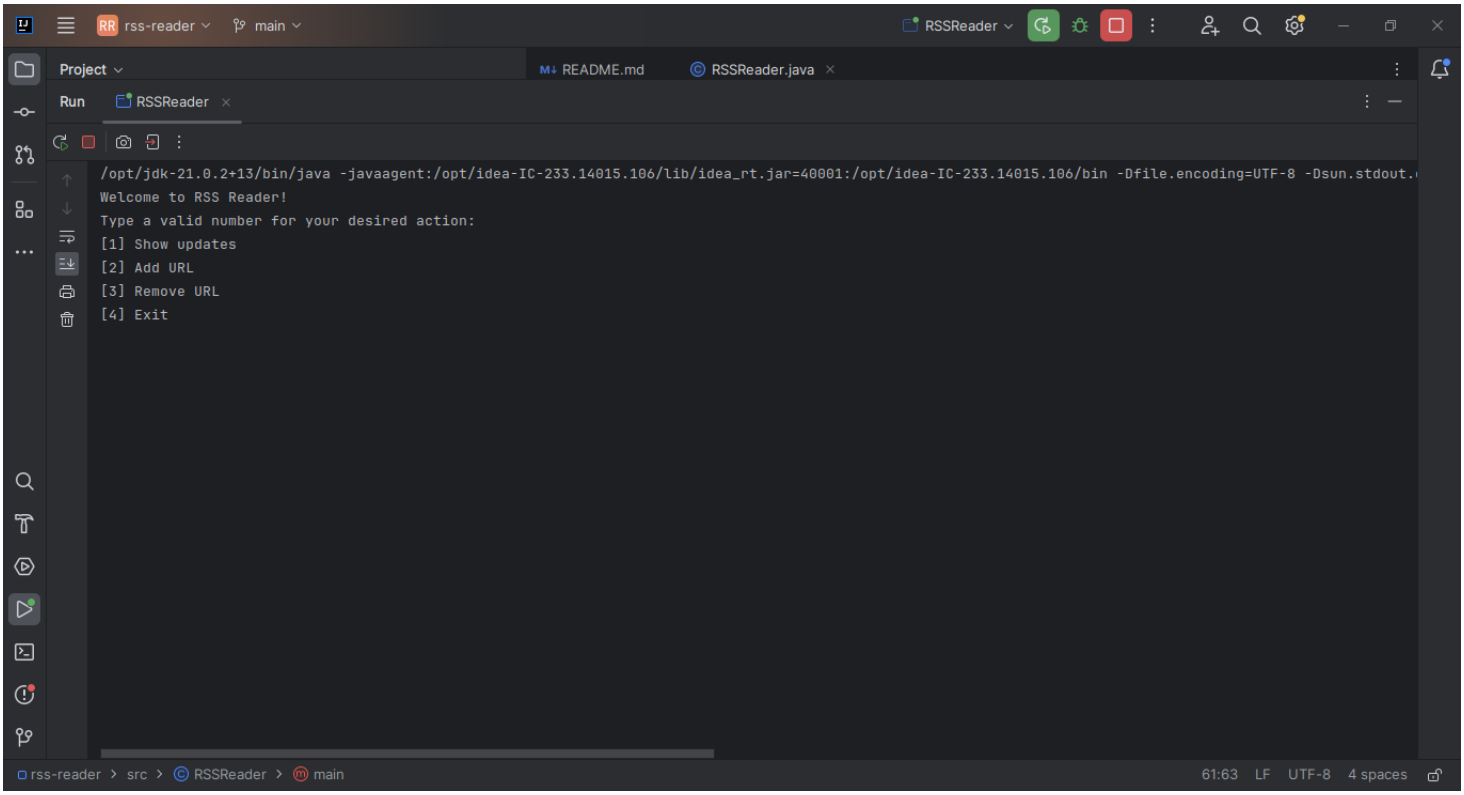
توجه داشته باشید که اگر در عنوان وبسایتی علامت ; وجود داشته باشد به مشکل برمی‌خوریم، اما در اینجا فرض می‌کنیم وبسایت‌هایی که با آنها کار می‌کنیم چنین نیستند.
برنامه‌ی مورد نظر ما چند قابلیت مهم خواهد داشت:

۱. اضافه کردن وبسایت‌ها در زمان اجرا و ثبت تغییرات پس از اتمام اجرای برنامه: باید قابلیت اضافه کردن وبسایت‌های جدیدی که می‌خواهیم دنبال کنیم را داشته و همچنین مطمئن باشیم که پس از اتمام اجرای برنامه و در شروع اجرای بعدی آن، اطلاعات جدید ثبت شده باشد.

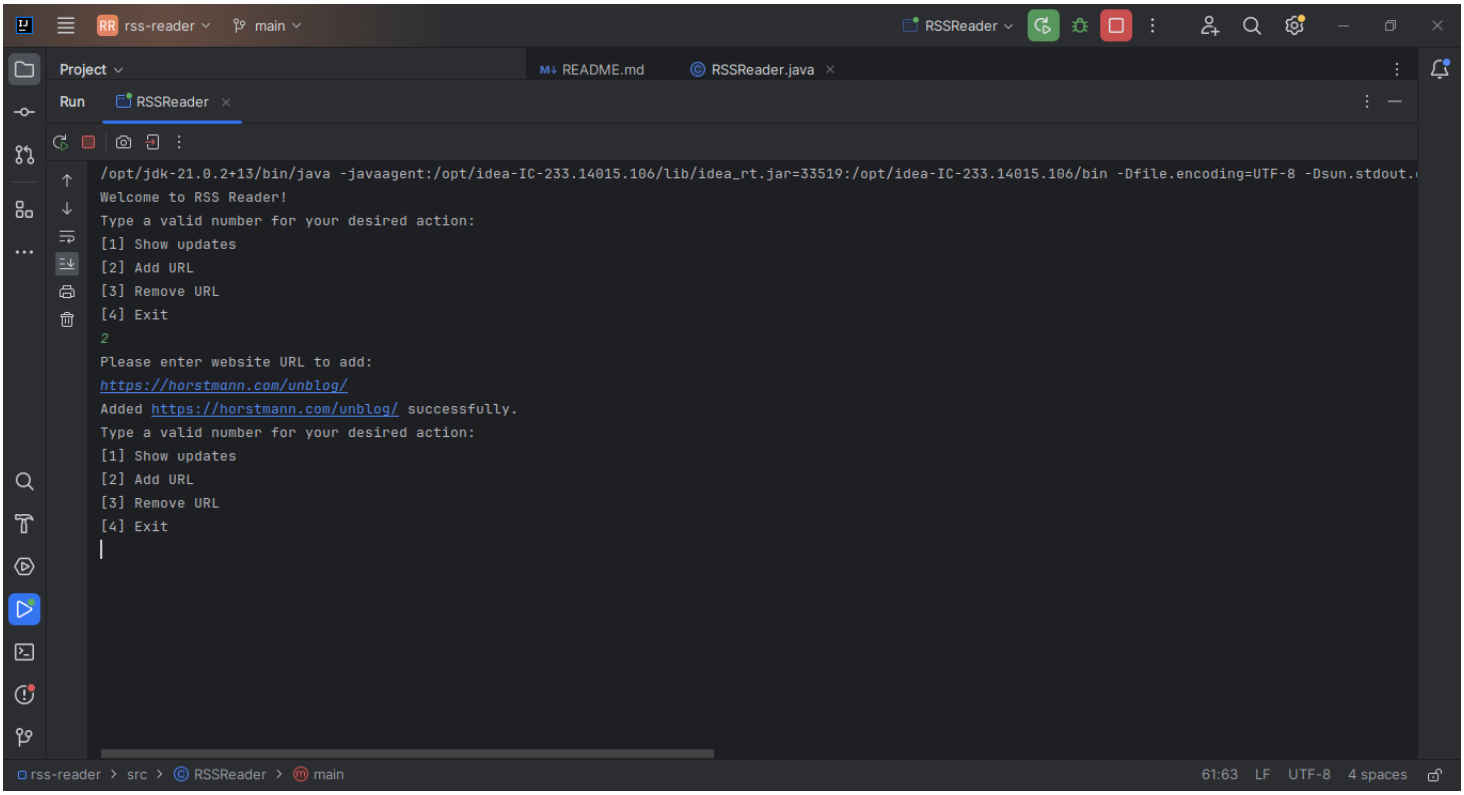
۲. حذف وبسایت‌ها در زمان اجرا و ثبت تغییرات پس از اتمام اجرای برنامه: ممکن است تعدادی از وبسایت‌هایی که قبلاً دنبال می‌کردیم را دیگر علاقه‌مند نباشیم که دنبال کنیم و باید قابلیت حذف آنها و ثبت این تغییرات در فایل را داشته باشیم.

۳. نمایش مطالب اخیر وبسایت‌ها: برنامه باید بتواند اطلاعات همه‌ی وبسایت‌های موجود داخل فایل علاوه بر آنهایی که در طول اجرای برنامه اضافه شده و بجز آنهایی که در طول اجرای برنامه حذف شده‌اند را به کاربر نمایش دهد. از آنجا که ممکن است مطالب اخیر وبسایت خیلی زیاد باشند می‌توان تعداد ثابتی از مطالب اخیر، مثلاً پنج تای اخیر را نمایش داد.

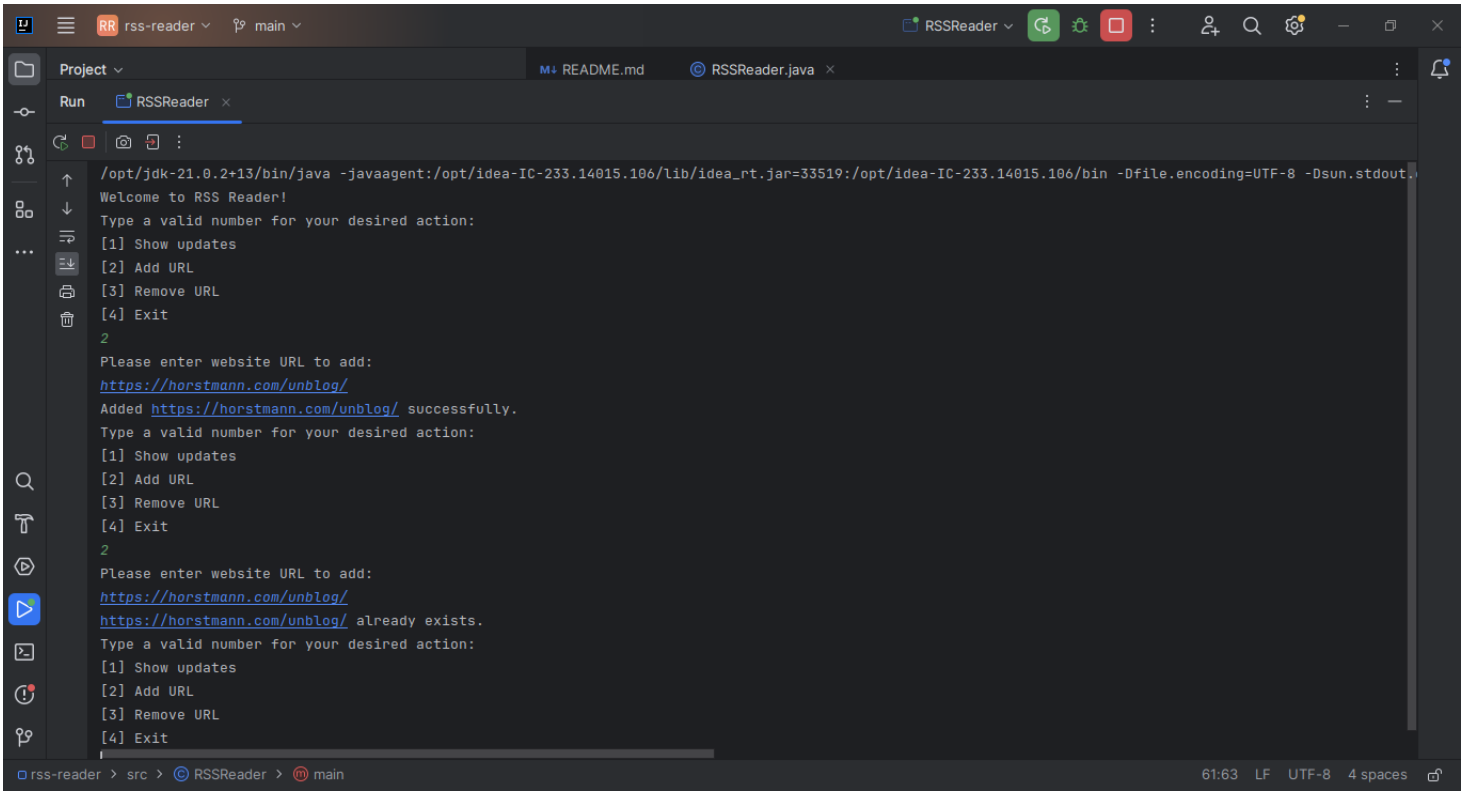
همان طور که گفته شد در پایان اجرای برنامه باید تغییراتی که در طول اجرای برنامه اتفاق افتاده در فایل data.txt ثبت شوند. همچنین تا زمانی که کاربر نخواسته باشد باید برنامه به اجرای خود ادامه داده و خطاهای احتمالی را تا حد ممکن هندل کرده و پیام خطا را نمایش دهد. در واقع انتظار داریم خطا و اشتباه در عملکرد یک متد باعث پایان ناگهانی کل برنامه نشود.
به یک بار اجرای برنامه‌ی نمونه‌ی RSS Reader توجه کنید:
در ابتدا این پیغام که در واقع راهنمای اولیه‌ی کاربر است نمایش داده می‌شود.



حال فرض کنید بخواهیم وبلاگ Cay Horstmann که قبلتر بررسی شد را اضافه کنیم. پس رقم 2 را تایپ کرده و پس از پیام چاپ شده نشانی وبلاگ را وارد می‌کنیم.

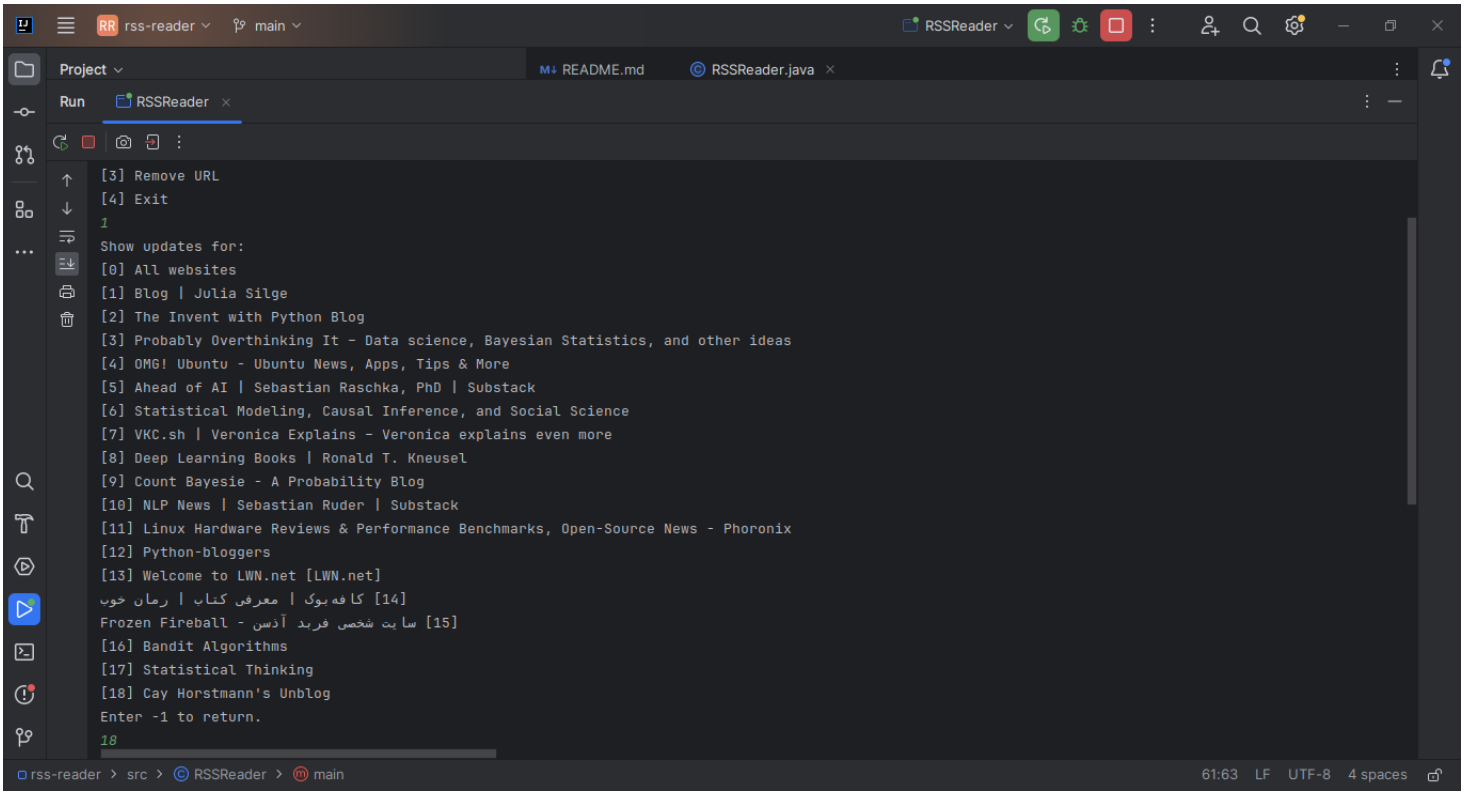


همان طور که مشاهده می‌کنید پیامی مبنی بر موفقیت اضافه شدن وبلاگ چاپ شده و دوباره راهنمای برنامه نمایش داده می‌شود. بیایید دوباره همین وبلاگ را اضافه کنیم تا ببینیم چه اتفاقی می‌افتد.

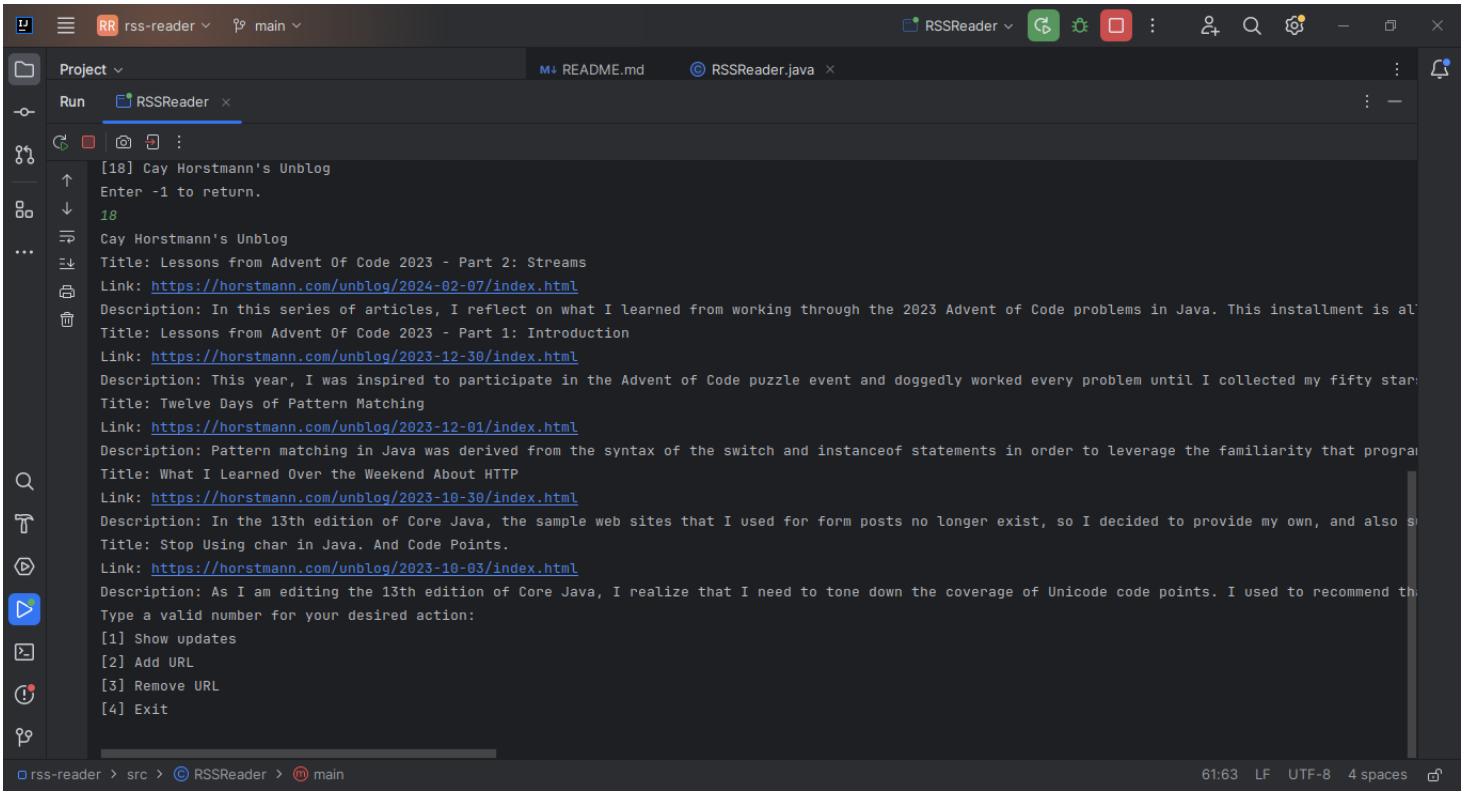


همان طور که ملاحظه می‌کنید پیغامی مبنی بر اینکه این آدرس از قبل وجود دارد نمایش داده می‌شود. انتظار می‌رود که برنامه‌ی شما هم حالات خاص این چنینی را به درستی هندل کند.

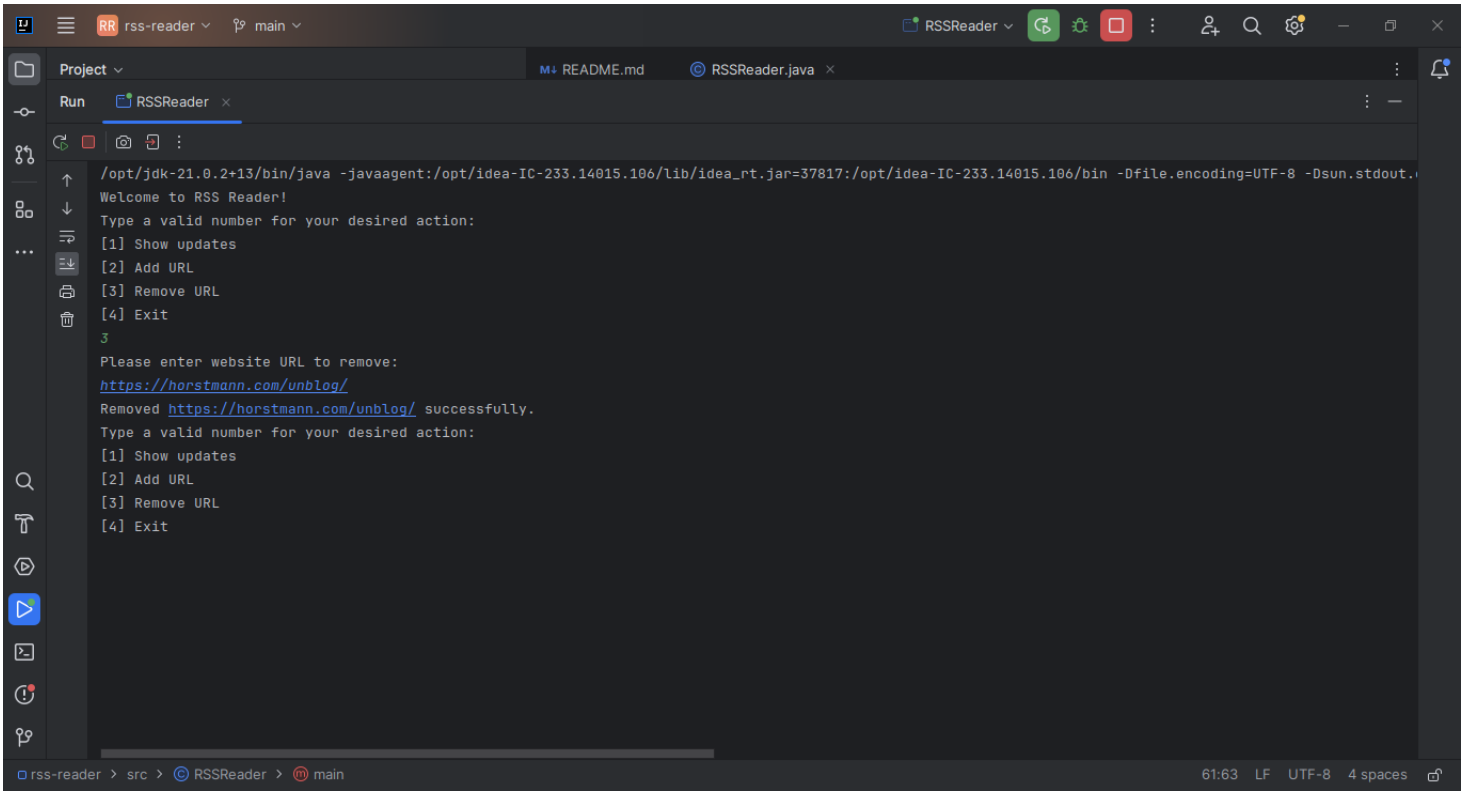
حال می‌خواهیم مطالب اخیر این وبلاگ را مشاهده کنیم؛ پس رقم 1 را وارد کرده و از بین وبسایت‌های وارد شده، عدد وبسایت مورد نظرمان را تایپ می‌کنیم.



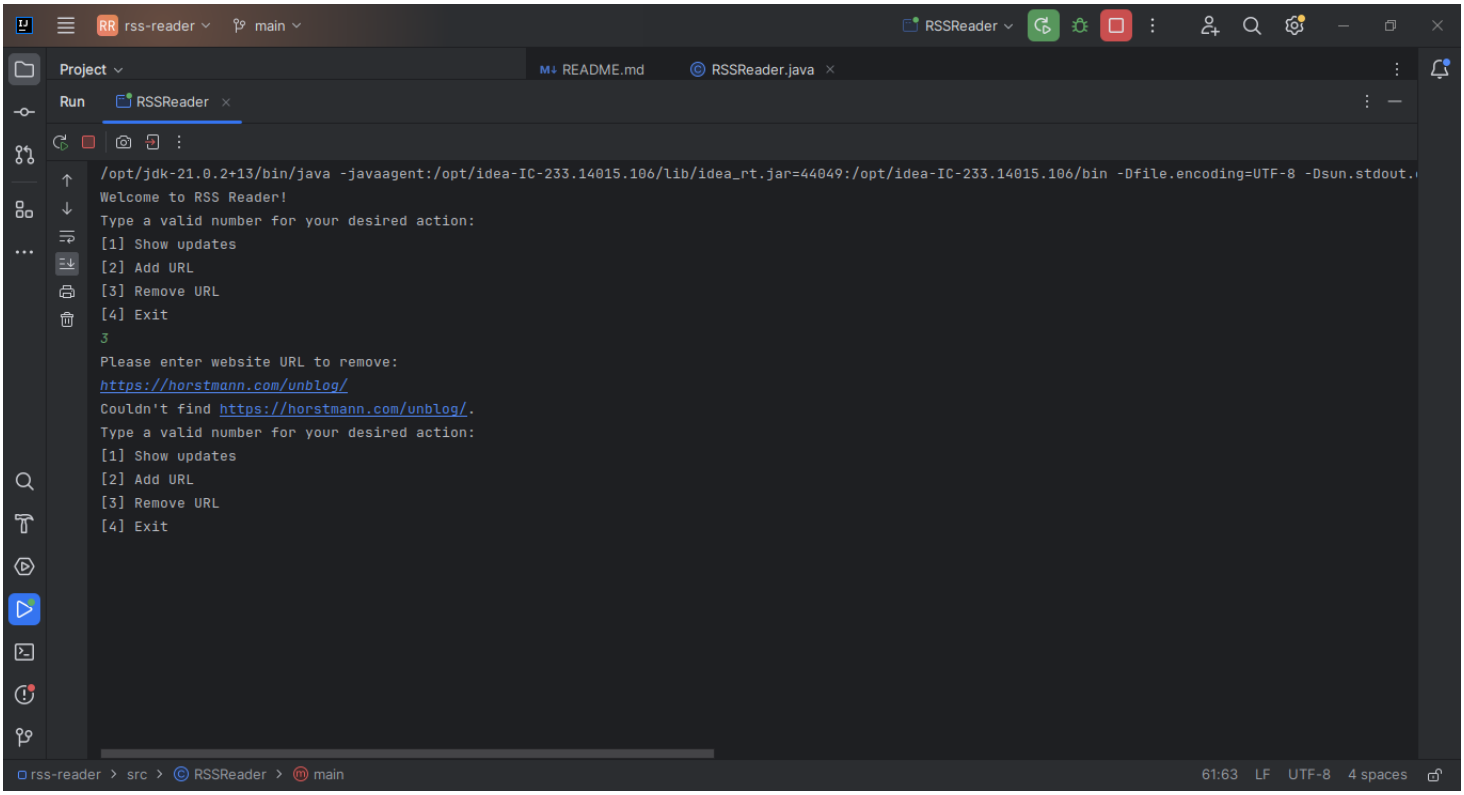
در ادامه عنوان پنج مطلب اخیر وبلاگ به همراه خلاصه و لینک آن نمایش داده می‌شود.



چنانچه به جای عدد 18 رقم 0 را وارد می‌کردیم پنج مطلب اخیر همه‌ی وبسایت‌هایی که دنبال می‌کنیم نمایش داده می‌شد و چنانچه 1- را وارد می‌کردیم به راهنما بازمی‌گشتیم. حال اگر بخواهیم وبلاگ را حذف کنیم رقم 3 و در ادامه آدرس وبلاگ را وارد می‌کنیم.



اگر بخواهیم دوباره این وبلاگ را حذف کنیم همان طور که در تصویر بعدی مشاهده می‌کنید با پیامی مبنی بر اینکه این وبلاگ یافت نشده مواجه می‌شویم.



برای پیاده‌سازی می‌توانید از این متدها استفاده کنید:

متد زیر برای به دست آوردن عنوان یک وبسایت از روی سورس htmlاش هست:

```java
public static String extractPageTitle(String html)
{
    try
    {
        org.jsoup.nodes.Document doc = Jsoup.parse(html);
        return doc.select("title").first().text();
    }
    catch (Exception e)
    {
        return "Error: no title tag found in page source!";
    }
}
```

این متد برای استخراج محتویات RSS از روی نشانی آن است:

```
public static void retrieveRssContent(String rssUrl)
{
    try {
        String rssXml = fetchPageSource(rssUrl);
        DocumentBuilderFactory documentBuilderFactory = DocumentBuilderFactory.newInstance();
        DocumentBuilder documentBuilder = documentBuilderFactory.newDocumentBuilder();
        StringBuilder xmlStringBuilder = new StringBuilder();
        xmlStringBuilder.append(rssXml);
        ByteArrayInputStream input = new ByteArrayInputStream(
                xmlStringBuilder.toString().getBytes("UTF-8"));
        org.w3c.dom.Document doc = documentBuilder.parse(input);
        NodeList itemNodes = doc.getElementsByTagName("item");

        for (int i = 0; i < MAX_ITEMS; ++i) {
            Node itemNode = itemNodes.item(i);
            if (itemNode.getNodeType() == Node.ELEMENT_NODE) {
                Element element = (Element) itemNode;
                System.out.println("Title: " + element.getElementsByTagName("title").item(0).getTextContent())
;
                System.out.println("Link: " + element.getElementsByTagName("link").item(0).getTextContent());
                System.out.println("Description: " + element.getElementsByTagName("description").item(0).
getTextContent());
            }
        }
    }
    catch (Exception e)
    {
        System.out.println("Error in retrieving RSS content for " + rssUrl + ": " + e.getMessage());
    }
}
```

با این متد می‌توانید نشانی RSS یک وبسایت را با داشتن url آن استخراج کنید:

```
public static String extractRssUrl(String url) throws IOException
{
    org.jsoup.nodes.Document doc = Jsoup.connect(url).get();
    return doc.select("[type='application/rss+xml']").attr("abs:href");
}
```

از این متد برای به دست آوردن سورس html یک وبسایت از روی url آن استفاده کنید:

```
public static String fetchPageSource(String urlString) throws Exception
{
    URI uri = new URI(urlString);
    URL url = uri.toURL();
    URLConnection urlConnection = url.openConnection();
    urlConnection.setRequestProperty("User-Agent", "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML,
 like Gecko) Chrome/108.0.0.0 Safari/537.36");
    return toString(urlConnection.getInputStream());
}
```

در متدهای فوق از متد کمکی زیر هم استفاده شده است:

```
private static String toString(InputStream inputStream) throws IOException
{
    BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(inputStream, "UTF-8"));
    String inputLine;
    StringBuilder stringBuilder = new StringBuilder();
    while ((inputLine = bufferedReader.readLine()) != null)
        stringBuilder.append(inputLine);

    return stringBuilder.toString();
}
```

معیارهای ارزیابی

۱. درصورتیکه فایلی برای ذخیره‌ی اطلاعات وبسایت‌ها وجود ندارد توسط برنامه ایجاد شده و در غیر این صورت فایل موجود در ابتدای برنامه بارگذاری شود. در پایان برنامه تغییراتی که در این راهنما به آنها اشاره شد در صورت اعمال توسط کاربر در داخل این فایل ذخیره شود.

۲. اطلاعات وبسایت‌ها باید مطابق فرمتی که گفته شد داخل فایل ذخیره شود.

۳. برای توابع و متغیرها نام‌های مناسب انتخاب کنید و کدهای خودتان را تا جایی که می‌توانید تمیز و واضح بنویسید و سعی کنید توابع را به گونه‌ای ایجاد کنید که هر تابع مخصوص یک عملکرد خاص باشد.

۴. ساختار برنامه مطابق آنچه که گفته شد باید رعایت شود. برنامه‌ی شما باید قابلیت اضافه کردن و حذف کردن وبسایت‌ها از داخل فایل و نشان دادن محتویات RSS آنها را داشته باشد.

۵. در هنگام بروز خطاهای احتمالی مثل حذف وبسایتی که وجود ندارد، خواندن محتویات فایل خالی، عدم دسترسی به اینترنت یا عدم دریافت نشانی RSS سالم و... مدیریت لازم را انجام داده و پیغام‌های مناسب چاپ کنید.

۶. حتماً برنامه را روی چند وبسایت مختلف امتحان کنید. برنامه‌ی شما در هنگام ارزیابی روی چند وبسایت امتحان خواهد شد.

۷. RSS وبسایت‌هایی که تگ link با type=”application/rss+xml” را دارند باید دریافت شود، ولی استخراج RSS مابقی وبسایت‌ها هدف این تمرین نیست. متد extractRssUrl که به شما داده شده فقط چنین RSSهایی را استخراج می‌کند. با این حال پیاده‌سازی استخراج RSS انواع دیگری از وبسایت‌ها هم پیشنهاد می‌شود.