



تمرین دوم: پارس کردن و نمایش فایل‌های XML

XML^۱ یک زبان نشانه‌گذاری ساده و استاندارد برای ذخیره‌سازی و مبادله اطلاعات به‌طور ساخت‌یافته است و به‌گونه‌ای طراحی شده که هم برای انسان و هم برای ماشین خوانش‌پذیر باشد. از آنجایی که این فرمت استاندارد در زمینه‌ها و ابعاد مختلفی کاربرد دارد، در این تمرین به پارس کردن^۲، استخراج و نمایش اطلاعات فایل‌های XML می‌پردازیم. روش‌های مختلفی برای پردازش XML وجود دارد که با سه‌تا از آن‌ها آشنا می‌شویم.

• SAX^۳

در این روش، پارسر روی فایل XML حرکت کرده و آن را به صورت برجسب به برجسب پیمایش می‌کند و به دلیل اینکه کل محتوای فایل به درون حافظه آورده نمی‌شود، برای زمانی که با فایل‌های نسبتاً بزرگ سروکار داشته باشیم مناسب است.

• DOM^۴

در این روش کل فایل XML در حافظه بارگذاری شده و یک ساختار درختی از آن ساخته می‌شود که ویرایش، حذف و اضافه کردن گره‌ها را ساده می‌کند.

• XSLT^۵

زبانی است که شیوه تبدیل یک فایل XML به فایلی دیگر را توصیف می‌کند و در آن تعیین می‌شود که چگونه ساختار درختی یک فایل XML به فایلی دیگر تبدیل شود. در اینجا از XSLT برای تبدیل XML به HTML و نمایش محتوای آن استفاده خواهیم کرد.

^۱ eXtensible Markup Language

^۲ Parsing

^۳ Simple API for XML

^۴ Document Object Model

^۵ eXtensible Stylesheet Language Transformations

فایل videogames.xml حاوی اطلاعاتی در مورد بازی‌های ویدیویی نظیر میزان فروش، امتیازها، ناشر، توسعه‌دهنده و غیره است. ساختار این فایل یک سری برچسب game برای هر بازی دارد:

```
<video_games>
  <game> ... </game>
  <game> ... </game>
  <game> ... </game>
  .
  .
  .
</video_games>
```

هر برچسب game یک id دارد و محتویات زیر را شامل می‌شود:

```
<game id=" ">
  <name> ... </name>
  <platform> ... </platform>
  <year_of_release> ... </year_of_release>
  <genre> ... </genre>
  <publisher> ... </publisher>
  <developer> ... </developer>
  <rating> ... </rating>
  <platform_type> ... </platform_type>
  <sales>
    <na_sales> ... </na_sales>
    <eu_sales> ... </eu_sales>
    <jp_sales> ... </jp_sales>
    <other_sales> ... </other_sales>
  </sales>
  <review_scores>
    <critic_score> ... </critic_score>
    <critic_count> ... </critic_count>
    <user_score> ... </user_score>
    <user_count> ... </user_count>
  </review_scores>
</game>
```

۱. برنامه‌ای بنویسید که این فایل را به روش SAX پارس کرده و خروجی‌ای دقیقاً به فرمت زیر برای بازی‌هایی که مقدار برچسب platform آن‌ها PC است تولید کند:

```
id;name;publisher;genre;year_of_release
```

خروجی باید براساس id مرتب باشد:

```
86;The Sims 3;Electronic Arts;Simulation;2009
139;World of Warcraft;Activision;Role-Playing;2004
193;Diablo III;Activision;Role-Playing;2012
219;StarCraft II: Wings of Liberty;Activision;Strategy;2010
285;Half-Life;Vivendi Games;Shooter;1997
.
.
.
```

۲. برنامه‌ای با استفاده از پارسر DOM بنویسید که نسخه‌ای جدید از فایل XML داده‌شده را تولید کند به‌طوری‌که:

- برچسب‌های rating و platform_type حذف شوند.
- برچسب sales میزان فروش یک بازی در آمریکای شمالی، اروپا، ژاپن و سایر مناطق را نشان می‌دهد. با جمع این مقادیر یک برچسب global_sales به جای sales به هر بازی اضافه شود. در صورت نیاز عدد خروجی را تا دو رقم اعشار گرد کنید.
- برچسب review_scores حاوی امتیازهای بررسی منتقدان و کاربران به هر بازی است و تعداد امتیازدهندگان برای هر دسته را هم شامل می‌شود. از این امتیازها یک میانگین وزن‌دار بگیرید:

$$\text{avg-score} = \frac{(\text{critic_score} \times \text{critic_count}) + (\text{user_score} \times \text{user_count})}{\text{critic_count} + \text{user_count}}$$

سپس براساس خروجی، یکی از عبارات زیر را به‌عنوان مقدار برای برچسب review_scores قرار دهید:

[0 , 20): very negative,
[20, 40): mostly negative,
[40, 60): mixed,
[60, 80): mostly positive,
[80, 100]: very positive

نمونه‌ای از این تغییرات گفته‌شده را در نمونه‌خروجی زیر مشاهده می‌کنید، بخش‌های قرمز رنگ حذف و بخش‌های سبز رنگ اضافه شده‌اند:

```
<game>
  <name> ... </name>
  <platform> ... </platform>
  <year_of_release> ... </year_of_release>
  <genre> ... </genre>
  <publisher> ... </publisher>
  <developer> ... </developer>
  <rating> ... </rating>
  <platform_type> ... </platform_type>
  <sales>
    <na_sales> ... </na_sales>
    <eu_sales> ... </eu_sales>
    <jp_sales> ... </jp_sales>
    <other_sales> ... </other_sales>
  </sales>
  <global_sales> ... </global_sales>
  <review_scores> ...
    <critic_score> ... </critic_score>
    <critic_count> ... </critic_count>
    <user_score> ... </user_score>
    <user_count> ... </user_count>
  </review_scores>
</game>
```

۳. فایل XML به دست آمده از مرحله قبل را به کمک XSLT به صورت HTML جدول بندی شده نمایش دهید به گونه ای که هر بازی و اطلاعات آن در یک سطر قرار بگیرد. مقدار border را برای جدول 1 قرار دهید تا کادربندی ساده ای هم داشته باشد. نمونه ای از این خروجی را در تصویر زیر می توانید ببینید:

Video Games

Name	Platform	Year of Release	Genre	Publisher	Developer	Global Sales	Review Scores
Wii Sports	Wii	2006	Sports	Nintendo	Nintendo	82.54	mostly positive
Mario Kart Wii	Wii	2008	Racing	Nintendo	Nintendo	35.52	very positive
Wii Sports Resort	Wii	2009	Sports	Nintendo	Nintendo	32.77	very positive
New Super Mario Bros.	DS	2006	Platform	Nintendo	Nintendo	29.8	very positive
Wii Play	Wii	2006	Misc	Nintendo	Nintendo	28.91	mostly positive
New Super Mario Bros. Wii	Wii	2009	Platform	Nintendo	Nintendo	28.32	very positive
Mario Kart DS	DS	2005	Racing	Nintendo	Nintendo	23.21	very positive
Wii Fit	Wii	2007	Sports	Nintendo	Nintendo	22.7	mostly positive
Kinect Adventures!	X360	2010	Misc	Microsoft Game Studios	Good Science Studio	21.82	mostly positive

-
- لطفاً تمرین را به صورت یک فایل فشرده حاوی کدها، خروجی ها و گزارش PDF با نام شماره دانشجویی خود به صورت StudentNumber.zip آپلود کنید.
 - خروجی تمرین باید sax.txt، dom.xml و dom.xsl باشد.
 - محدودیتی در استفاده از زبان های برنامه نویسی ندارید.
 - در گزارشی که نوشته می شود به طور مختصر زبان برنامه نویسی مورد استفاده و نحوه اجرای کدها را توضیح داده و نحوه انجام هریک از بخش های تمرین را شرح دهید. در ابتدای گزارش نام و شماره دانشجویی حتماً نوشته شود.

در صورت هرگونه ابهام یا سؤال در مورد تمرین به reza.gharibi@live.com ایمیل بزنید.