ИСПОЛЬЗОВАНИЕ ВЫЧИСЛЕНИЙ НА СТОРОНЕ КЛИЕНТА ПРИ СОЗДАНИИ ВЕБ-СЕРВИСОВ ОБРАБОТКИ ПРОСТРАНСТВЕННЫХ ДАННЫХ, НА ПРИМЕРЕ ТЕХНОЛОГИИ JAVA WEB START

Е.А. Паниди, Д.С. Ефимов**
**ФГБОУ ВО Санкт-Петербургский государственный университет*
Санкт-Петербург, Россия, panidi@yandex.ru

USING OF THE CLIENT-SIDE COMPUTATIONS WHEN IMPLEMENTING WEB SERVICES FOR SPATIAL DATA PROCESSING, THE CASE STUDY OF JAVA WEB START TECHNOLOGY

E.A. Panidi, D.S. Efimov**
**Saint-Petersburg State University*
Saint-Petersburg, Russia, panidi@yandex.ru

Abstract. One of the key areas of Web geotechnology development is the implementation of software tools and systems, which are capable not only to display geospatial data in the Web interface, but also to provide functionality for processing and analysis directly in the browser window. Significant feature of current Web-based geospatial standards is the focusing on server-side data processing. Our study investigates possibilities and general ways of decentralized data processing implementation on the client side in the case of using the Java Web Start technology. The test software tools are developed that implements capability of transmitting the executable program code to the client computer through the Web interface and spatial data processing on the client side.

Keywords: Web-GIS, Client-Side Web Geoprocessing, Java Web Start.

Введение. Создание программных средств и систем способных не только отображать пространственные данные в веб-интерфейсе, но и предоставлять пользователям инструменты для обработки и анализа непосредственно в окне браузера, является одним из ключевых направлений развития веб-геотехнологий (т.е. веб-технологий, предназначенных для манипулирования пространственными данными и их представления в веб-среде). Значимой особенностью современных стандартов использования пространственных данных в глобальной сети является ориентированность на сервер-стороннюю обработку данных. В исследованиях авторов данной статьи рассматриваются возможности и пути внедрения децентрализованной обработки данных на стороне клиента на примере технологии JavaWebStart. Создан тестовый пример программного обеспечения, реализующего возможности передачи исполняемого программного кода на клиентский компьютер через веб-интерфейс и выполнения обработки пространственных данных на клиентской стороне.

Постановка проблемы. В последние годы активное развитие получили технологии и решения, связанные с представлением пространственных данных, и в частности материалов дистанционного зондирования (ДЗ), в глобальной сети. Развитие подобных технологий началось с представления пространственных данных в веб-интерфейсе, но достаточно быстро продвинулось к пониманию потребности в разработке технологий интерактивной работы с пространственными данными в веб-среде. Таким образом, технологии веб-картографии эволюционировали в веб-ГИС-технологии.

В целом, задачи, решаемые в веб-ГИС, можно разделить на задачи представления данных (поиск, конвертация, трансформирование данных) и задачи обработки (пространственный анализ, генерация производных данных). В обоих случаях основной парадигмой проектирования и разработки веб-картографических и веб-ГИС решений сегодня является централизованная клиент-серверная архитектура (облачная архитектура [5] в случае массивных проектов, ориентированных на большие данные) которая подразумевает выполнение основных вычислений и обработки данных на стороне сервера. В ряде случаев, такой подход становится неприемлем. Например, в ситуации, когда значительный объём исходных данных хранится на стороне клиента, и может быть дорого или неудобно передавать их на сервер при повторяющейся обработке. Также препятствием могут стать узкие или нестабильные каналы связи. Наконец, если пользователь оперирует данными, имеющими коммерческие или законодательные ограничения, то передача их на сторонний сервер или в публичное облако может быть не приемлемой [4]. Заметим, что указанные особенности вполне актуальны в отечественных условиях при работе с материалами ДЗ.

Возможный выход из данной ситуации, который чаще всего применяется в организациях – создание частной вычислительной инфраструктуры или использование частного облака, что требует существенных затрат. В случае ограниченности ресурсов выходом могли бы стать использование вычислительных мощностей клиента для обработки данных и передача по сети не данных, а программных инструментов для их обработки. Такой подход позволит сохранить преимущества модели SaaS [14], заключающиеся в централизованном администрировании программных средств, на мощностях сервера, и дополнить их возможностью построения более устойчивых распределённых систем [1– 3].

Однако, как было упомянуто выше, современные веб-ГИС используют преимущественно серверстороннюю обработку. Технологии промышленного уровня для исполнения программного кода на стороне

клиента (Adobe Flash[15], Microsoft Silverlight[16], JavaScript[17], Java[18]) имеют давнюю историю и достаточно разнообразны, но в рамках веб-ГИС технологий они используются преимущественно для представления данных (например, OpenLayers framework[19], ArcGIS Viewer for Silverlight[20], ArcGIS Web App Builder[21]). Более того, единственный международный стандарт, формализующий требования к функциональности веб-сервисов обработки пространственных данных – OGC WPS (Open Geospatial Consortium Web Processing Service), предусматривает только сервер-стороннюю архитектуру веб-сервисов [13].

Технологии обработки данных на стороне клиента используются лишь в отдельных проектах, но по мнению авторов данной статьи, выработка стандартных подходов к созданию инструментов обработки данных в веб-среде, основанных на клиент-сторонних вычислениях, могла бы дать большую гибкость в проектировании и реализации веб-ГИС, особенно при исполнении мелкомасштабных проектов и в научных исследованиях. Целесообразность такого взгляда подтверждается, в том числе, упоминаниями других авторов о перспективности проведения разработок в данном направлении. Так Steven Keens, в контексте использования и развития стандарта WPS, упоминает идею мобильного кода, т.е. кода, который может быть передан для запуска с сервера на клиентский компьютер [9]. Также, предварительное описание архитектуры сервисов обработки пространственных данных, при создании сетевой инфраструктуры обработки материалов дистанционного зондирования, приводят Yves Coene и др. [6]. Тем не менее, комплексных решений вопросов разработки технологий и стандартов передачи исполняемого программного кода с сервера на клиент при обработке пространственных данных в веб-ГИС сегодня не существует. В данной статье описаны некоторые результаты, полученные авторами в процессе апробации технологии Java Web Start[22] при создании веб-приложений для обработки пространственных данных.

Материалы и методы. Технология Java Web Start (Java WS) является частью семейства технологий разработки и развёртывания программных средств, в основе которых лежит язык программирования Java [Gosling at al., 2014]. Данная технология позволяет передавать исполняемый код приложения с сервера на клиент и осуществлять запуск приложения на клиентском оборудовании. Приложения, развёртываемые с использованием Java WS, в общем случае, не требуют специализированной установки на клиентский компьютер и каких-либо специфических знаний от пользователя.

Технология Java WS имеет несколько ключевых преимуществ, делающих её полностью готовой для создания ГИС-приложений и развёртывания их по модели SaaS:

• Так как программы, созданные на языке Java, исполняются в специализированном программном окружении – на платформе JavaStandardEdition (Java SE), а именно, внутри среды исполнения Java Runtime Environment (JRE), то это делает программный код платформенно независимым. Единожды созданный и опубликованный на сервере программный код может исполняться практически на любом клиенте, в том числе, в операционных системах семейства Windows, начиная с Windows 98 и старше. Среда исполнения, в данном случае хоть и требует отдельной установки, как правило, уже присутствует на клиентском компьютере, благодаря широкому распространению семейства технологий Java в целом.

• Технология Java WS поддерживает несколько версий платформы Java SE, и во многих случаях веб-приложения Java WS совместимы сразу с несколькими версиями платформы. В ситуациях, когда те или иные компоненты приложения не поддерживают версию Java SE, установленную на клиенте, приложение использует встроенные средства проверки совместимости и самостоятельно оповещает о необходимости использования корректной версии клиентской платформы, а также автоматически загружает и устанавливает необходимую версию.

• В составе технологии Java WS реализованы встроенные средства безопасности, в частности, использование механизма подписывания программного кода, который позволяет пользователю убедиться в том, что программный код получен из доверенного источника.

• Java WS позволяет осуществлять запуск приложений как из окна браузера (при этом само приложение запускается вне браузера), так и в автономном режиме, без использования браузера и без подключения к сети, если запуск из браузера представляется неудобным или невозможным. В последнем случае подразумевается, что приложение, ранее уже запускалось в режиме онлайн и было автоматически локально кэшировано.

В целом, Java WS представляет собой вспомогательную технологию развёртывания веб-приложений. Сами приложения состоят из набора файлов формата JAR, содержащих исполняемый код приложения на языке Java и все дополнительные ресурсы приложения (файлы настроек и т.д.). Дополнительные

[15] https://www.adobe.com/products/flashruntimes.html
[16] http://www.microsoft.com/silverlight/
[17] http://en.wikibooks.org/wiki/JavaScript
[18] http://www.java.com
[19] http://openlayers.org
[20] http://www.esri.com/software/arcgis/viewer-for-silverlight
[21] http://www.esri.com/software/web-appbuilder
[22] http://www.oracle.com/technetwork/java/javase/javawebstart/index.html

ресурсы могут быть получены и в явном виде по прямой веб-ссылке, но в подобном случае не будет происходит их добавление в локальный кэш приложения на клиентском компьютере.

Особо следует отметить, что технология JavaWS занимает промежуточное положение между технологиями JavaServlet/JavaPortlet и JavaApplet [7]. Спецификации JavaServlet и JavaPortlet регулируют порядок развёртывания сервер-сторонних веб-приложений, спецификация JavaApplet – порядок развёртывания клиент-сторонних веб-приложений, исполняемых в окне веб-браузера. Две первых технологии позволяют создавать полноценные приложения для обработки данных и обеспечивать доступ к этим приложениям через веб-интерфейс, однако все операции с данными производятся исключительно на стороне сервера. В случае с технологией JavaApplet, операции производятся на стороне клиента, а само приложение запускается как компонент веб-браузера.

Основным способом запуска приложения Java WS, как было упомянуто выше, является запуск непосредственно из окна браузера, в результате перехода по веб-ссылке, указывающей на файл в формате JNLP. Файл JNLP, это текстовый файл, содержание которого структурировано на языке разметки XML. Данный файл содержит общее описание приложения Java WS и ссылки на программные компоненты, необходимые для запуска. Именно с автоматического перехвата ссылки на файл JNLP и его открытия в JavaWebStartLauncher (приложение в составе JavaSE) и начинается запуск приложения Java WS.

В результате считывания файла JNLP происходит автоматическая загрузка на клиентский компьютер необходимых для работы приложения файлов (файлов JAR), если они не были загружены ранее, их добавление в локальный кэш и собственно запуск приложения.

Для того, чтобы предупредить ситуацию, при которой компоненты Java WS не установлены на клиентском компьютере в код веб-страницы необходимо добавлять скрипт, предназначенный для проверки наличия компонент Java WS в клиентской операционной системе и их автоматической установки, при необходимости.

Таким образом, весь процесс запуска веб-приложения осуществляется полностью автоматически, за исключением первого перехода по ссылке для открытия JNLP файла. После успешного выполнения первого запуска, появляется возможность запуска приложения уже из кэша или с рабочего стола при помощи ярлыка. Ярлыки на рабочем столе и(или) в меню Пуск могут создаваться автоматически, если компоненты Java настроены соответствующим образом. В простейшем случае, пользователь может сохранить сам файл JNLP локально и запускать его повторно уже без использования браузера.

Также возможен запуск приложения как из кэша, так и с сервера с использованием командной строки. В таком случае, в командную строку вводится команда *javaws jnlp_url*, в которой вместо *jnlp_url* указывается веб-ссылка или локальный путь к файлу JNLP.

Подпись кода для приложенийJava WS, требующих доступ к локальному диску, обязательна и является одним из средств защиты от вредоносного кода. Если проверка подписи у таких приложений по какой-либо причине не сработает, то приложение не будет запущенно. При запуске таких приложений на клиенте отображается диалоговое окно, содержащее информацию о том, кем подписан код. Пользователь, в свою очередь, убедившись, подписан ли код издателем (разработчиком) заслуживающим доверия, может принять решение о разрешении или запрете запуска. Приложение будет запущено только в том случае, если пользователь подтвердит, что доверяет издателю.

Файлы JAR могут быть подписаны разработчиком как самостоятельно, путём генерации специализированных ключей, так и с помощью доверенного сертификата, который можно выпустить платно в специализированном центре сертификации. В последнем случае центр сертификации становится сторонним гарантом подлинности цифровой подписи и обеспечивает возможность проверки подписи в режиме онлайн. В последних версиях Java, не подписанные приложения Java WS и приложения, подписанные ключом, который разработчик сгенерировал самостоятельно, по умолчанию не запускаются на клиентском компьютере, а их файлы автоматически удаляются. В связи с этим, подпись с помощью доверенного сертификата технически наиболее удобна, так как в данном случае на стороне клиента не возникает проблем с запуском приложения.

В случае работы с приложением, подписанным разработчиком самостоятельно, конечный пользователь должен убедиться в том, что приложение загружается с сервера заслуживающего доверия, и добавить этот веб-сервер в исключения безопасности Java. Данная операция не проста и заключается в том, что пользователь должен внести адрес веб-сервера в список исключений, в окне свойств Java на клиентском компьютере (рис. 1). При использовании корректно настроенных приложений JavaWS других дополнительных настроек не требуется.

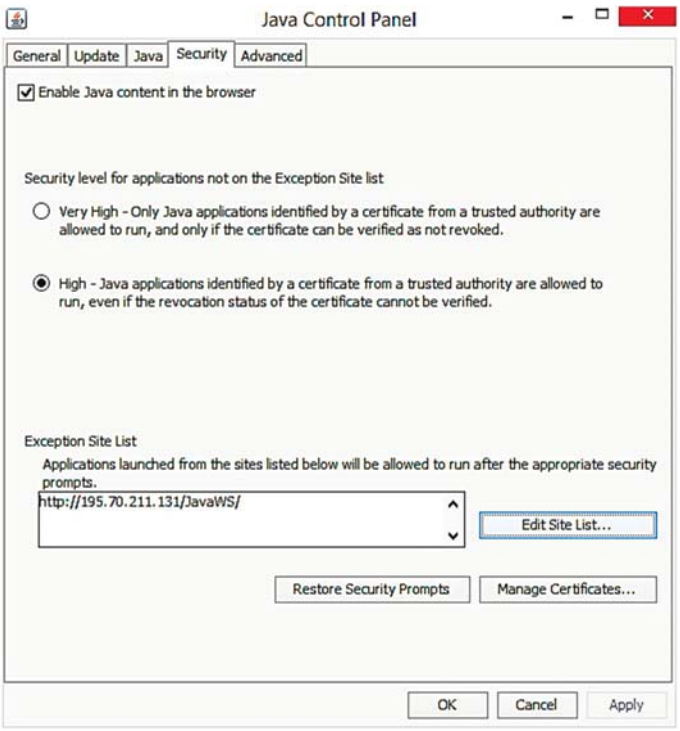

Рис. 1. Добавление веб-сервера в список исключений Java. Вкладка Security в окне свойств Java – Java Control Panel

Структура файла JNLP, как и структура любого файла, написанного на языке разметки XML, сформирована набором стандартных тегов (элементов) и хранимых в них значений. Код такого файла имеет следующий вид:

```
<?xmlversion='1.0'encoding='UTF-8'?>
<jnlp spec='1.0+' codebase='195.70.211.131/JavaWS' href='Landsat_DN_conversions.jnlp'>
        <information>
                <title>Landsat DN/Radiance/Reflectance/Temperature Conversion</title>
                <vendor>SPSU</vendor>
                <offline-allowed/>
        </information>
        <update check='always'/>
        <security>
                <all-permissions/>
        </security>
        <resources>
                <j2se version='1.8+'/>
                <jar href='Landsat_DN_conversions.jar' main='true'/>
        </resources>
        <application-desc main-class='DN_conversions'></application-desc>
</jnlp>
```

В приведённом примере представлены необходимые тэги файла JNLP. Корневым тэгом является *<jnlp>*, он может включать в себя атрибуты *spec*, *codebase*, *href* и *version*. В атрибуте *spec* указывается минимальная версия спецификации JNLP, которую должны поддерживать компоненты Java на клиентском компьютере. Атрибут *href* содержит относительную ссылку на файл JNLP, а атрибут *codebase* – веб-ссылку, которая используется в качестве базовой для всех относительных ссылок на компоненты приложения, в том числе для ссылки в атрибуте *href*.Атрибут *version* определяет версию запускаемого приложения и является не обязательным, в приведённом примере он не указан.

Остальные тэги являются дочерними для *<jnlp>*, они разделяются на несколько групп: информационные тэги, тэги, описывающие состав ресурсов и параметры обновления, тэги, описывающие параметры безопасности, а также тэги, описывающие параметры самого файла JNLP.

Группа информационных тэгов описывается внутри тэга *<information>*. Данная группа включает тэги *<title>*, отвечающий за название приложения, и *<vendor>*, содержащий наименование разработчика приложения. Также в этой группе может присутствовать тэг, отвечающий за развёртывание приложения

<*offline-allowed*>. При отсутствии данного тэга приложение будет невозможно запустить без сетевого подключения.

За обновления отвечает тэг <*update*>, который, в свою очередь, содержит атрибут *check*, указывающий на то, когда приложение будет проходить проверку на наличие обновлений.

Тэг безопасности <*security*> отвечает за включение неограниченного доступа к ресурсам клиентского компьютера. Получение такого доступа происходит при объявлении тэга <*all-permission*> внутри <*security*> и при наличии подписи у всех файлов JAR.

Для указания ссылок на все ресурсы, используемые приложением, используется тег <*resources*>. Он включает в себя тэги <*jar*> и <*j2se*>. Первый определяет имя файла JAR, которое является частью пути к классам приложения, а второй – версию среды исполнения Java, необходимую для запуска приложения.

Последний необходимый тэг, это <*application-desc*>. Он указывает на то, как файл JNLP должен запускать приложение. В приведённом примере, это запуск в качестве настольного приложения. Для запуска может потребоваться указание имени главного класса приложения, которое выполняется в атрибуте *main-class*.

Результаты. Для апробации технологии JavaWS при создании веб-ГИС, было создано приложение (разработка программного кода выполнена Д.С. Ефимовым), позволяющее осуществлять конвертацию целочисленных значений яркости каналов космических снимков Landsat (так называемых, значений Digital-Numbers, или DN-значений) в значения излучения, принятого на сенсор (Radiance), значения отражательной способности земной поверхности (Reflectance) и значения температуры, вычисляемые на основе тепловых инфракрасных каналов.

Указанная разработка была использована в качестве тестового примера, так как подразумевает выполнение наиболее типичных операций, связанных с обработкой пространственных данных, каких как собственно вычисления с использованием нелинейных зависимостей, чтение и запись текстовых, графических и простых бинарных файлов, выполнение операций над данными, представленными в различных форматах. Разработанное приложение предназначено для выполнения обработки данных съёмки Landsat 7, которые включают в себя 8-битные каналы мультиспектральной съёмки (растровые слои) и метаданные (текстовые файлы).

Разработанное приложение выполняет следующие операции:

• Конвертация значений пикселей снимка в показатели излучения, принятого на сенсор, фотометрическую величину, характеризующую величину излучения, отражённого от земной поверхности и дошедшего до спутника;

• Конвертация в величину отражательной способности земной поверхности без учёта влияния атмосферы;

• Конвертация данных тепловых инфракрасных каналов в значения температуры.

Входными данными для обработки служат непосредственно 8-битные растровые файлы в формате TIFF и текстовые файлы метаданных, поставляемые вместе со снимками. Результат обработки может быть сохранён в виде в виде простого бинарного файла.

Вычисление требуемых значений пикселей выполняется по стандартным формулам, известным из технической документации к данным съёмки Landsat 7 (NationalAeronauticsandSpaceAdministration, 2010). Согласно документации, конвертации в величину излучения, величину отражательной способности и значения температуры должны проходить по следующим формулам, соответственно:

$$L_\lambda = \frac{(LMAX_\lambda - LMIN_\lambda) \cdot \left(QCAL - QCAL_{min}\right)}{QCAL_{max} - QCAL_{min}} + LMIN_\lambda \tag{1}$$

$$\rho_p = \frac{\pi \cdot L_\lambda \cdot d^2}{ESUN_\lambda \cdot cos\theta_S} \tag{2}$$

$$T = \frac{K2}{ln\left(\frac{K1}{L_\lambda} + 1\right)} \tag{3}$$

где: Lλ – величина излучения (Ватт/м^2·ср·мкм); 〖LMAX〗_λ и 〖LMIN〗_λ – соответственно максимальное и минимальное значения излучения, зафиксированного сенсором для данного канала; 〖QCAL〗_max и 〖QCAL〗_min соответственно максимальное и минимальное целочисленны е значения пикселей канала; QCAL – целочисленное значение пикселя канала, для которого необходимо вычислить значение величины излучения; ρ_p – отражательная способность; π – число ПИ; d – расстояние от Земли до Солнца в астрономических единицах; 〖ESUN〗_λ-средняя интенсивность солнечного излучения для данного канала (Ватт/м^2·мкм); θ_S – зенитный угол Солнца в градусах; T – температура в кельвинах; K1, K2 – калибровочные коэффициенты.

Взаимодействие с пользователем реализуется с помощью графического пользовательского интерфейса. Запуск отдельного метода или класса приложения из командной строки, в данном случае не возможен, хотя технология JavaWS подобный запуск допускает. При этом, однако, требуется выделять все функции в отдельные JAR файлы.

Приложение реализует 7 операций (методов), необходимых для обработки снимков (рис. 2):

1. Загрузка изображения для просмотра;
2. Загрузка метаданных;
3. Выбор канала для обработки
4. Конвертация в величину излучения в соответствии с выбранным каналом;
5. Конвертация в величину отражательной способности в соответствии с выбранным каналом;
6. Конвертация в значения температуры в соответствии с выбранным каналом;
7. Сохранение рассчитанных значений в файл либо в виде уменьшенной копии в графический файл в формате JPG, для предварительного просмотра.

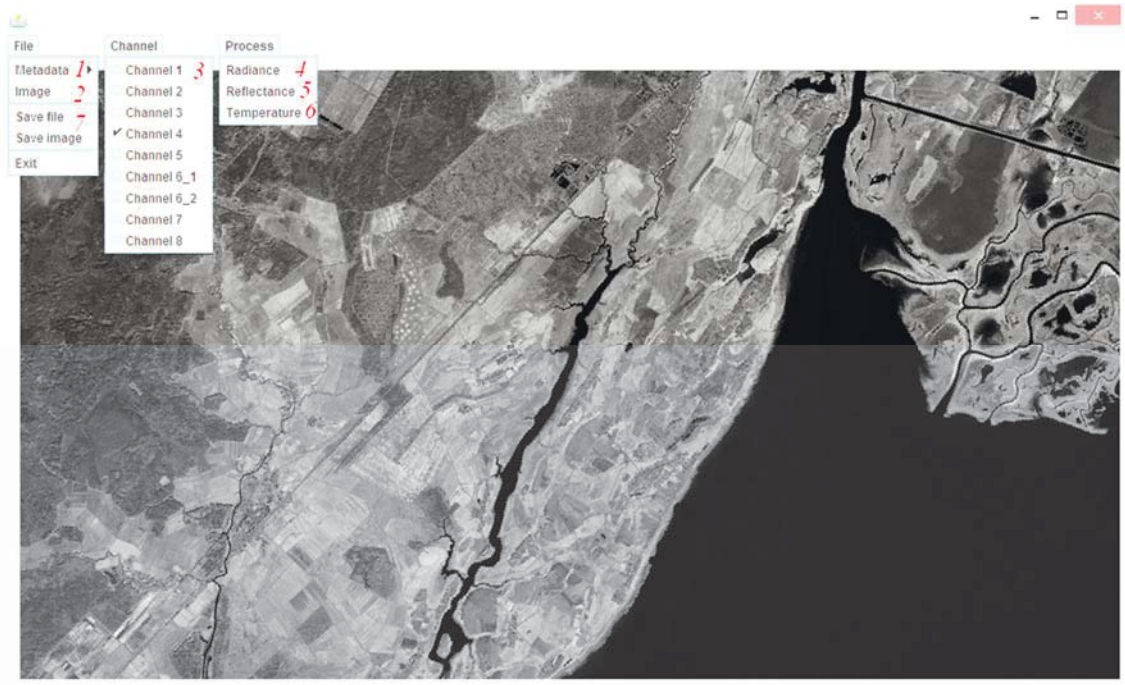

Рис. 2. Графический пользовательский интерфейс созданного приложения для обработки снимков Landsat

Первый метод позволяет указать на локальном диске файл в формате TIFF, содержащий данные одного из каналов съёмки Landsat. Данная операция предназначена для просмотра данных и позволяет открыть, в том числе, изображение в формате TIFF, не являющееся спутниковым снимком.

Второй метод реализует загрузку метаданных. Он позволяет открыть текстовый файл метаданных, который поставляется вместе со снимками, и считать его содержимое в текстовую переменную, с последующим поиском в ней всех необходимых для расчётов констант.

Третий метод позволяет выбрать для последующей обработки один из каналов снимка. Список каналов при этом формируется на основе данных, считанных из файла метаданных.

Методы 4–6 позволяют запустить вычисление тех или иных требуемых значений. Наконец, последний из основных методов осуществляет сохранение полученного массива значений в простой бинарный файл. Также, при необходимости, пользователь может сохранить уменьшенную копию обработанного изображения (канала съёмки) в графический файл в формате JPG, для предварительного просмотра.

Выводы. В результате проделанных работ выделены основные преимущества технологии JavaWS при создании веб-приложений, такие как: возможность запуска в различных операционных системах, работа с разными версиями платформы Java, возможность запуска в изолированной программной среде, возможность запускать приложение автономно. Авторами был рассмотрен механизм работы данной технологии и различные способы запуска приложений, созданных на её основе.

Тестовая разработка выполнена на примере приложения, осуществляющего конвертацию целочисленных значений яркости каналов космических снимков Landsat в значения излучения, принятого на сенсор и значения отражательной способности земной поверхности. Результат разработки опубликован и доступен по ссылке *http://195.70.211.131/JavaWS*.

Результаты, полученные при проведении тестовой разработки, демонстрируют, что технология JavaWS позволяет обеспечить эффективную разработку и веб-публикацию приложений для обработки пространственных данных по модели SaaS. На основе выполненной тестовой разработки можно реализовать

более сложные и комплексные ГИС-приложения, а также веб-сервисы для обработки пространственных данных.

Исследование выполнено при поддержке РФФИ в рамках научного проекта № 13-05-12079 офи_м.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК
REFERENCES

1. *Казаков Э.Э.* Подходы к созданию веб-сервисов обработки и анализа материалов дистанционного зондирования. // Дистанционное зондирование Земли из космоса: алгоритмы, технологии, данные. Материалы молодёжной школы-семинара (г. Барнаул, 2–6 октября 2013 г.). Барнаул, Азбука, 2013. С. 82–86.

Kazakov E.E. Approaches to Implementation of Web Services for Processing and Analysis of Remote Sensing Data. Proceedings of the Earth Remote Sensing From Space: Algorithms, Technology, Data – Young Scientists Workshops (Altai State University, Barnaul, Russia, October 2–6, 2013). Barnaul, Azbuka, 2013. pp. 82–86. (In Russian, accessible at http://elibrary.asu.ru/xmlui/bitstream/handle/asu/203/read.7book?sequence=1).

2. *Казаков Э.Э., Капралов Е.Г., Паниди Е.А., Терехов А.В.* Геосервисы как функциональная основа геопортала: опыт и перспективы. // Обработка пространственных данных и дистанционный мониторинг природной среды и масштабных антропогенных процессов (DPRS'2013). Избранные труды конференции. (30 сентября – 04 октября 2013 г. Барнаул). Барнаул, Пять плюс, 2013. С. 14–23.

Kazakov E.E., Kapralov E.G., Panidi E.A., Terekhov A.V. Geoservices as the Functional Basis of the Geoportal: Experience and Prospects. Processing of the Conference. Spatial Data and Remote Monitoring of the Natural Environment and Large-Scale Anthropogenic Processes (DPRS'2013), September 30 – October 04, 2013, Barnaul. Barnaul, Five Plus, 2013. pp. 14–23.(In Russian, accessible at http://www.iwep.ru/ru/bibl/books/matkonf/2_DPRS_stati_A5.pdf).

3. *Паниди Е.А.* Геосервисы для онлайн-обработки данных дистанционного зондирования. // Дистанционное зондирование Земли из космоса: алгоритмы, технологии, данные. Материалы молодёжной школы-семинара (г. Барнаул, 2–6 октября 2013 г.). Барнаул, Азбука, 2013. С. 74–81.

Panidi E.A. Geoservices for Online Remote Sensing Data Processing. Proceedings of the Earth Remote Sensing From Space: Algorithms, Technology, Data – Young Scientists Workshops (Altai State University, Barnaul, Russia, October 2–6, 2013). Barnaul, Azbuka, 2013. pp. 74–81. (In Russian, accessible at http://elibrary.asu.ru/xmlui/bitstream/handle/asu/203/read.7book?sequence=1).

4. *Паниди Е.А.* Подходы к разработке клиентосторонних веб-сервисов геообработки. // Устойчивое развитие территорий: картографо-геоинформационное обеспечение. Материалы Международной конференции, Белгород, Харьков (Украина), Кигали (Руанда) и Найроби (Кения), 23 июля – 8 августа 2014 г. С. 205–214.

Panidi E.A. Approaches to Development of Client-Side Geoprocessing Web-Services. InterCarto/InterGIS-20: Sustainable development of territories: Cartography and GI Support. Proceedings of the International Conference, Belgorod (Russia), Kharkov (Ukraine), Kigali (Rwanda) and Nairobi (Kenya), July 23 – August 8, 2014. pp. 205–214. (In Russian , accessible at http://intercarto.ru/data/ic2014.pdf).

5. Buyya R., Broberg J., Goscinski A. (eds.) Cloud Computing: Principles and Paradigms. John Wiley & Sons, Inc. p. 637, 2011. DOI: 10.1002/9780470940105.fmatter

6. Coene Y., Marchetti P.G., Smolders S. Architecture and Standards for a Distributed Digital Library of Geospatial Services. // Third Italian Research Conference on Digital Library Systems (IRCDL 2007) Padova 29–30 January 2007 Proceedings. 2007. pp. 52–60.

7. Gosling J., Joy B., Steele G., Bracha G., Buckley A. The Java language specification. Addison-Wesley Professional, 2014. 792 p.

8. Kazakov E., Terekhov A., Kapralov E., and Panidi E.: WPS-based technology for client-side remote sensing data processing, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XL-7/W3, 643–649, doi: 10.5194/isprsarchives-XL-7-W3-643-2015, 2015.

9. Keens S. (ed.) Discussions, findings, and use of WPS in OWS-4. OGC Discussion Paper. OGC 06-182r1. Version 0.9.1, 2007-05-10. 2007.

10. Landsat 7 Science Data Users Handbook. National Aeronautics and Space Administration, 2010. 186 p.

11. Mueller M., Pross B. (eds.) OGC WPS 2.0 Interface Standard. OGC Implementation Standard. OGC 14-065. Version 2.0, 2015-03-05. 2015.

12. Panidi E.A. Towards Client-Side Web Processing Services. Proceedings of theOSGeo's European Conference on Free and Open Source Software for Geospatial, Independent Innovation for INSPIRE, Big Data and Citizen Participation (FOSS4G-Europe 2014) July 15–17 2014 Jacobs University, Bremen, Germany, p. 4, 2014. Issued on-line, accessible at http://foss4g-e.org/content/toward-client-side-web-processing-services

13. Schut P. (ed.) OpenGIS Web Processing Service. OpenGIS Standard. OGC 05-007r7. Version 1.0.0, 2007-06-08. 2007.

14. Software as a Service: Strategic Backgrounder. Software & Information Industry Association, Washington, D.C., 28 February 2001.