

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**РАЗРАБОТКА ВЕБ-АРІ ЭЛЕКТРОННОЙ БИБЛИОТЕКИ СГУ ДЛЯ
СИСТЕМЫ ДИСТАНЦИОННОГО ОБУЧЕНИЯ IPSILONUNI**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы

направления 02.03.02 — Фундаментальная информатика и информационные
технологии

факультета КНиИТ

Шепелева Дениса Дмитриевича

Научный руководитель

доцент, к. ф.-м. н.

А. С. Иванова

Заведующий кафедрой

к. ф.-м. н., доцент

С. В. Миронов

Саратов 2021

ВВЕДЕНИЕ

С каждым днем эра информационных технологий набирает все новые обороты. Не секрет, что разработки в этой сфере призваны, в первую очередь, облегчить и улучшить нашу повседневную жизнь. На сегодняшний день многие учебные заведения нашей страны начинают процесс интеграции современных технологий в традиционные подходы к обучению. Наш университет не является исключением — на базе СГУ уже успешно функционируют портал дистанционного обучения IpsilonUni и электронная библиотека, с которыми ежедневно работают наши преподаватели и студенты. Ранее упомянутые ресурсы непрерывно связаны между собой и опираются друг на друга при составлении учебного плана — списка источников (литературы). В связи с этим, необходимо сделать этот процесс максимально простым и комфортным для заинтересованных в этом пользователей — преподавателей. Отсюда, целью настоящей работы является разработка веб-API электронной библиотеки СГУ для системы дистанционного обучения IpsilonUni.

В процессе выполнения работы были поставлены следующие задачи:

- рассмотреть теоретическую базу, необходимую для выполнения настоящей работы;
- ознакомиться со спецификациями электронной библиотеки СГУ и IpsilonUni;
- разработать техническое задание и, опираясь на него, реализовать веб-API.

Бакалаврская работа состоит из введения, 2-х разделов, заключения, списка использованных источников и приложения. Общий объем работы — 46 страниц, из них 40 страниц — основного текста. Список использованных источников состоит из 31 наименования.

Первый раздел «Теоретическая часть» содержит подробное описание теории, необходимой для реализации практической части работы. Во втором разделе «Практическая часть» рассмотрен общий процесс разработки, с которым довелось столкнуться для достижения поставленной цели: планирование ожидаемых результатов, возникшие проблемы и варианты их решения, конечная реализация.

1 Теоретическая часть

В теоретической части рассмотрены основные понятия, определения и технологии, необходимые для реализации практической части настоящей работы.

1.1 Цикл разработки программного обеспечения

Разработчику крайне важно знать и понимать процесс разработки ПО для того, чтобы во время работы не теряться в хаосе происходящего. От проекта к проекту этот процесс может отличаться, но в общем случае можно выделить следующие этапы:

1. идея — поиск и выбор исполнителя;
2. анализ — формализация требований;
3. дизайн — выбор архитектуры и технологий;
4. разработка — написание кода, разработка интерфейса и тому подобное;
5. тестирование — проверка на соответствие требованиям;
6. выкладка — выпуск на рынок;
7. поддержка — работа с пользователями.

Модель разработки ПО подразумевает способ комбинирования выше упомянутых пунктов, определяя выбор стратегии, расписание, необходимые ресурсы. Классическими моделями считаются [1]:

- водопадная (каскадная);
- V-образная;
- итерационная инкрементная;
- спиральная;
- гибкая.

1.1.1 Водопадная (каскадная)

Каскадная модель [1] — Одна из самых старых моделей, подразумевающая последовательное прохождение стадий, каждая из которых должна завершиться полностью до начала следующей. Водопадная модель позволяет легко управлять проектом благодаря ее жесткости. Тем не менее, водопадная модель часто интуитивно применяется при выполнении относительно простых задач, а ее недостатки послужили отправным пунктом для создания новых моделей.

1.1.2 V-образная

V-образная модель [2], является логическим развитием водопадной. В общем случае как водопадная, так и V-образная модели жизненного цикла ПО могут содержать один и тот же набор стадий, но принципиальное отличие заключается в том, как эта информация используется в процессе реализации проекта. Данная модель позволяет минимизировать риски, а также обнаружить и устранить множество потенциальных проблем до того, как они станут проблемами реальными.

1.1.3 Итерационная инкрементная

Итерационная инкрементальная модель [3] является фундаментальной основой современного подхода к разработке ПО. Как следует из названия модели, ей свойственна определенная двойственность:

- С точки зрения жизненного цикла, модель является итерационной, так как подразумевает многократное повторение одних и тех же стадий;
- С точки зрения развития продукта (приращения его полезных функций), модель является инкрементальной.

Ключевой особенностью данной модели является разбиение проекта на относительно небольшие промежутки (итерации), каждый из которых в общем случае может включать в себя все классические стадии, присущие водопадной и v-образной моделям. Итогом итерации является приращение (инкремент) функциональности продукта, выраженное в промежуточной сборке (билде).

1.1.4 Спиральная

Спиральная модель [4] представляет собой частный случай итерационной инкрементальной модели, в котором особое внимание уделяется управлению рисками, в особенности влияющими на организацию процесса разработки проекта и контрольные точки. Стоит обратить внимание на то, что здесь явно выделены четыре ключевые фазы:

- проработка целей, альтернатив и ограничений;
- анализ рисков и прототипирование;
- разработка (промежуточной версии) продукта;
- планирование следующего цикла.

1.1.5 Гибкая

Рассмотрев все предыдущие модели, можно перейти к последней — гибкой [5]. Данная модель представляет собой совокупность различных подходов к разработке ПО и базируется на так называемом «agile-манифесте» [6]:

- люди и взаимодействие важнее процессов и инструментов;
- работающий продукт важнее исчерпывающей документации;
- сотрудничество с заказчиком важнее согласования условий контракта;
- готовность к изменениям важнее следования первоначальному плану.

Иными словами, гибкая модель представляет собой облегченную, с точки зрения документации, смесь итерационной инкрементальной и спиральной моделей. Данная модель, основанная на agile-манифесте, являлась наиболее подходящей в рамках решения поставленных в работе задач, в связи с чем и была выбрана в качестве основной.

1.2 Веб-API

Веб-API [7] — интерфейс прикладного программирования для веб-сервера или веб-браузера. Благодаря веб-API, программные компоненты могут взаимодействовать друг с другом, в следствие чего подобное решение позволяет разработчику использовать готовые блоки для построения приложения. Как правило, компоненты образуют некоторую иерархию — компоненты, расположенные на верхних уровнях, используют API более низкоуровневых компонентов. По подобному принципу построены протоколы передачи данных по Интернету.

1.2.1 Веб-API ASP.NET

Веб-API ASP.NET [8] — платформа, позволяющая легко создавать HTTP-службы для широкого диапазона клиентов, включая браузеры и мобильные устройства. Данная платформа работает на базе .NET Framework [9] и заточена для разработки в REST-стиле [10]. REST-архитектура, схематически изображенная на рисунке ??, предполагает применение следующих типов HTTP-запросов к серверу:

- GET — получение записей;
- POST — добавление записей;
- PUT — редактирование записей;
- DELETE — удаление записей.

ASP.NET имеет некоторые особенности, в следствие чего и был выбран в качестве основополагающей платформы настоящей работы:

- позволяет работать на большинстве популярных языков программирования, в том числе и на С# [11];
- компилируемый код выполняется быстрее, а большинство ошибок устраняется еще на стадии разработки;
- расширяемый набор элементов управления и библиотек классов, ускоряющий разработку;
- расширяемые модели событий, обработки запросов и серверных элементов управления.

1.3 Продукты семейства ИРБИС

Продукты семейства ИРБИС [12] неразрывно связаны с электронной библиотекой СГУ. ИРБИС [13] — система автоматизации библиотек, разработанная международной ассоциацией пользователей и разработчиков электронных библиотек и новых информационных технологий. Семейство ИРБИС включает в себя такие продукты, как:

- ИРБИС64 — интегрированная система автоматизации в составе TCP/IP сервера баз данных и семи модулей: Комплектатор, Каталогизатор, Читатель, Книговыдача, Администратор, Книгообеспеченность, Корректор;
- ИРБИС64+ — система автоматизации, предназначенная для создания и ведения электронной библиотеки;
- ИРБИС128 — многоплатформенное решение на основе Web-технологии и распределенных БД, имеющее трехзвенную архитектуру (СУБД, сервер приложений, клиент);
- Web-ИРБИС — шлюз для доступа к базам данных ИРБИС по Web-технологии;
- J-ИРБИС 2.0 — модуль для создания библиотечных порталов и электронных библиотечных систем;
- ИРБИС Имидж-каталог — система для создания имидж-каталогов на основе распознавания сканированных образов каталожных карточек;
- ИРБИС-Аналитика — система полностью автоматического заимствования аналитических описаний журнальных статей;
- ИРБИС-АРМ центр регистрации читателей — обеспечение защиты персональных данных в системе ИРБИС;
- Z-ИРБИС — модуль доступа к ресурсам ИРБИС64 через Интернет по

протоколу Z39.50 [14], включая средства обеспечения работы в режиме Web-сервиса;

- СК-ИРБИС — средство для создания корпоративных (сводных) каталогов.

1.3.1 Система ИРБИС64+

ИРБИС64+ [15] представляет собой систему автоматизации [16], предназначенную для создания и ведения электронной библиотеки, в отличие от ИРБИС64, который предназначен для ведения электронного каталога. ИРБИС64+ позволяет осуществлять:

- полнотекстовый поиск с использованием критерия релевантности на основе автоматического разбиения текстов на страницы и их пословной индексации;
- представление результатов поиска в порядке убывания релевантности документов, а в рамках одного документа – в порядке убывания релевантности найденных страниц;
- постраничный просмотр полных текстов с обеспечением маркировки найденных слов запроса и навигации по релевантным страницам, оглавлению и личным закладкам.

Информационной основой ИРБИС64+ являются базы данных, представляющие собой совокупность связанных библиографических данных и полных текстов изданий. Библиографические данные ведутся на основе коммуникативного формата RUSMARC [17], а в качестве полных текстов используются распознанные PDF-файлы, которые подвергаются автоматическому разбиению на страницы и индексированию по словам.

1.4 Портал IpsilonUni

IpsilonUni [18] — образовательный портал, система дистанционного обучения, разработанная СГУ на базе технологий HTML Slim и Ruby on Rails. IpsilonUni позволяет:

- работать с учебными планами и образовательной траекторией обучающегося;
- фиксировать ход образовательного процесса (использовать балльно-рейтинговую систему);
- работать с учебно-методическими курсами и тестами;

- формировать портфолио обучающегося.

1.4.1 HTML Slim

HTML Slim (шаблонизатор Slim) [19] — программное обеспечение, позволяющее использовать html-шаблоны для генерации конечных html-страниц. Основная цель Slim — сократить объем кода, при этом оставляя его легко поддерживаемым. По своей сути, это обычный HTML [20], но без $\langle \rangle$ скобок, в котором используются отступы для обозначения вложенных элементов.

1.4.2 Ruby on Rails

Коробочные системы управления сайтами не подходят, если разрабатывается что-то отличное от стандартного сайта или если к проекту предъявляются действительно высокие требования к быстродействию и устойчивости к нагрузкам. В этом случае в качестве платформы для разработки выбирается фреймворк Ruby on Rails (RoR) [21] — фреймворк, написанный на языке программирования Ruby, реализующий архитектурный шаблон Model-View-Controller и использующий REST-стиль построения для веб-приложений. Основными компонентами являются:

- модель (model) — предоставляет остальным компонентам приложения объектно-ориентированное отображение данных. Объекты модели могут осуществлять загрузку и сохранение данных в реляционной базе данных, а также реализуют бизнес-логику;
- представление (view) — создает пользовательский интерфейс с использованием полученных от контроллера данных. Представление также передает запросы пользователя на манипуляцию данными в контроллер;
- контроллер (controller) — набор логики, запускаемой после получения HTTP-запроса сервером.

2 Практическая часть

В практической части настоящей работы рассматривается общий процесс разработки.

2.1 Постановка задачи

Как сделать процесс добавления списка источников, который используется при составлении рабочей программы в IpsilonUni, максимально комфортным и как можно менее трудозатратным для преподавателей (далее — пользователей)? Чтобы ответить на этот вопрос, в первую очередь, необходимо было изучить сам процесс составления ранее упомянутой программы. На тот момент, интересующий этап выглядел следующим образом, где пользователь:

- открывает сайт IpsilonUni;
- заполняет основную информацию о рабочей программе;
- открывает сайт электронной библиотеки СГУ;
- осуществляет поисковой запрос по интересующим его книгам в электронной библиотеке;
- если книги найдены с определенным количеством свободных экземпляров, то они добавляются в рабочую программу через IpsilonUni вручную.

Очевидно, что выше проиллюстрированный процесс был неудобен для пользователя, так как включал в себя несколько промежуточных звеньев, которые могли бы быть автоматизированы. В этом и заключалась задача — продумать, как можно исключить из процесса формирования списка источников шаги, подразумевающие прямое взаимодействия пользователя с библиотекой.

2.2 Изучение спецификаций

Этот этап играл важнейшую роль на протяжении всей разработки веб-API. Необходимо было очень тщательно изучить все доступные документы, чтобы подобрать подходящую стратегию решения задачи. Именно здесь и пришлось столкнуться с большинством проблем.

2.2.1 Электронная библиотека СГУ

Самый универсальный путь для изучения спецификаций — интервью с заказчиками (в данном случае — с сотрудниками библиотеки). По завершении цикла бесед было выяснено, что электронная библиотека СГУ не имела собственного программного обеспечения и использовала готовое решение в виде

системы ИРБИС64+. На сегодняшний день, большая часть продуктов семейства ИРБИС представляет из себя достаточно архаичное программное обеспечение с очень специфичной и сложной для понимания рядового разработчика архитектурой. Здесь и появились первые проблемы:

- отсутствие единой, общей спецификации;
- как следствие первого, отсутствие аккумулирующей (накапливающей в себе) системы спецификаций — приходилось буквально искать и собирать требуемую информацию с разных ресурсов по частям;
- некоторые спецификации частично дублировали друг друга, что еще больше усложняло процесс поиска интересующей информации.

2.2.2 Портал IpsilonUni

IpsilonUni разрабатывался для внутренних нужд университета и к моменту написания настоящей работы не успел обзавестись документацией. Всю интересующую информацию предоставлял ведущий разработчик портала — Карякин Денис Алексеевич. Работу усложняли еще такие факты, как:

- IpsilonUni мог быть установлен только под управлением операционной системы семейства Linux [22];
- портал использовал такие технологии как HTML Slim и Ruby on Rails, в которых на момент написания настоящей работы отсутствовал практический опыт разработки.

2.3 Стратегия решения задачи

После изучения и анализа всей доступной документации были отдельно вынесены следующие тезисы:

- система ИРБИС64+ может быть установлена только под управлением операционной системы Windows, в то время как IpsilonUni — под управлением операционной системы семейства Linux;
- модули ИРБИС64+ либо не предоставляли открытого исходного кода, либо имели слишком сложную для понимания спецификацию.

В ходе очередного поиска информации по системе ИРБИС64+, на официальном форуме была найдена библиотека ManagedIrbis [23], написанная на языке C#. Однако, как и в случае с самой системой, здесь отсутствовала какая-либо адекватная документация. Таким образом, было принято решение написать веб-API на языке C#, которое должно было принимать HTTP-запросы

с IpsilonUni, получать и производить парсинг данных с баз ИРБИС64+, переводить данные в интересующий формат (например, JSON) и возвращать их порталу в качестве ответа.

2.4 Реализация веб-API

2.4.1 Первые результаты

Удалось подключиться к локально установленному серверу ИРБИС64+ и получить тестовые данные всех книг, у которых автор начинался на букву «А». Однако, полученные результаты отличались от ожидаемых — они представляли собой набор полей с определенным набором других подполей в зависимости от их типа. Эта проблема в очередной раз поставила разработку в тупик и заставила обратиться за помощью на форум ИРБИС. Именно там была найдена ссылка на таблицу с интересующей информацией [24]. Спецификация полностью описывала полученные ранее результаты: у поля с главным тегом есть несколько подполей с другими тегами, которые могут быть отображены в виде символьного или буквенного представления. Несмотря на частичный выход из сложившейся ситуации, появилась новая проблема — какие именно поля необходимо получать из базы.

2.4.2 Уточнение технического задания

После серии уточняющих бесед с преподавателями и сотрудниками библиотеки, было решено отложить вопрос конкретизации полей на более поздние этапы разработки. Это было связано во многом из-за временных рамок, а также из-за отсутствия технического понимания стороны вопроса (сотрудники библиотеки не могли знать системные номера интересующих тегов, которые они заполняли с помощью графического интерфейса). В конечном итоге, задача была сведена к следующему:

- получить из базы такие поля, как: автор, заглавие, издательство, ссылка;
- добавить возможность поиска по автору и заглавию книги;
- вывести результирующие данные в JSON-формате.

2.4.3 Архитектура веб-API

Веб-API представляет из себя контроллер и определенный набор моделей (сущностей).

Класс `Publisher` (далее — Издатель) получается путем наследования класса `IrbisField`, который будет рассмотрен позднее. Опираясь на ранее упомянутую табличную спецификацию, за Издателя отвечает поле с тегом 210. Здесь имеется несколько подполей:

- `Year` — год издательства;
- `Name` — название издателя;
- `City` — город издательства.

Все значения получаются путем запроса определенных подполей с тегами «d», «с», «а» соответственно. Остальные модели конструируются по образцу и подобию только что рассмотренной.

`IrbisField` является родителем всех моделей. Здесь определена переменная, отвечающая за тег поля, а также методы конвертации значений полей из внутреннего представления системы ИРБИС64+.

Класс `Book` — результирующая модель, которая должна быть получена на выходе. Она состоит из других моделей: Заголовка, Издателя, Информации об онлайн-ресурсе.

Теперь, можно перейти к контроллеру, который отвечает за функционал выдачи книг по запросу пользователя. Осуществляется подключение пользователя к базе, поисковой запрос, выдача результатов.

2.4.4 Тестирование запросов

Стоит заранее подчеркнуть, что тестирование производится локально, в связи с чем все компоненты должны быть установлены на одном и том же компьютере.

В самом начале требуется запустить ранее установленный сервер ИРБИС64+. Теперь, с помощью средств `IIS Express` [25], предоставляемых средой разработки `Visual Studio` [26], необходимо запустить реализованное ранее веб-API. После запуска, откроется новое окно веб-браузера, в адресной строке которого необходимо ввести поисковой запрос. В качестве первого теста следует передать корректные данные, например «`/api/books?searchType=0&query=Кентавр&count=999`».

ЗАКЛЮЧЕНИЕ

В ходе выполнения настоящей работы была проведена большая исследовательская деятельность, где на практике были получены навыки обращения с технической документацией, анализа, общения с заказчиками, реальный опыт командной веб-разработки. Совокупность всех полученных в ходе работы навыков позволила реализовать промежуточную версию веб-API, которая, в дальнейшем, будет являться связующим звеном между пользователем, электронной библиотекой СГУ и системой дистанционного обучения IpsilonUni. Также, здесь стоит выразить огромную благодарность всем заинтересованным лицам, которые принимали активное участие в процессе разработки.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 What are the Software Development Models? [Электронный ресурс].— URL: <https://tinyurl.com/y9lmx9k> (Дата обращения 05.05.2021). Загл. с экр. Яз. англ.
- 2 What is V-model- advantages, disadvantages and when to use it? [Электронный ресурс].— URL: <https://tinyurl.com/ycbcv7jc> (Дата обращения 05.05.2021). Загл. с экр. Яз. англ.
- 3 Куликов, С. С. Тестирование программного обеспечения / С. С. Куликов. — Минск: Четыре четверти, 2017.
- 4 Spiral model [Электронный ресурс].— URL: <https://tinyurl.com/yc9p4gp5> (Дата обращения 05.05.2021). Загл. с экр. Яз. англ.
- 5 What is Agile model—advantages, disadvantages and when to use it? [Электронный ресурс].— URL: <https://tinyurl.com/y7bsyue5> (Дата обращения 05.05.2021). Загл. с экр. Яз. англ.
- 6 Manifesto for Agile Software Development [Электронный ресурс].— URL: <https://www.agilealliance.org/agile101/the-agile-manifesto/> (Дата обращения 01.06.2021). Загл. с экр. Яз. англ.
- 7 Web APIs [Электронный ресурс].— URL: <https://developer.mozilla.org/en-US/docs/Web/API> (Дата обращения 01.05.2021). Загл. с экр. Яз. англ.
- 8 Обзор ASP.NET [Электронный ресурс].— URL: <https://docs.microsoft.com/ru-ru/aspnet/overview> (Дата обращения 01.06.2021). Загл. с экр. Яз. рус.
- 9 Общие сведения о платформе .NET [Электронный ресурс].— URL: <https://docs.microsoft.com/ru-ru/dotnet/framework/get-started/overview> (Дата обращения 01.06.2021). Загл. с экр. Яз. рус.
- 10 What is Rest [Электронный ресурс].— URL: <https://restfulapi.net/> (Дата обращения 05.05.2021). Загл. с экр. Яз. англ.

- 11 C# documentation [Электронный ресурс]. — URL: <https://docs.microsoft.com/en-us/dotnet/csharp/> (Дата обращения 10.05.2021). Загл. с экр. Яз. англ.
- 12 *Бродовский, А. С.* Семейство продуктов ирбис: Современное состояние и перспективы развития / А. С. Бродовский // *НАУЧНЫЕ И ТЕХНИЧЕСКИЕ БИБЛИОТЕКИ*. — 2015. — Vol. 11. — Pp. 80–82.
- 13 Продукты семейства ИРБИС [Электронный ресурс]. — URL: <https://tinyurl.com/yj3tdpy2> (Дата обращения 03.05.2021). Загл. с экр. Яз. рус.
- 14 Gateway to Library Catalogs [Электронный ресурс]. — URL: <http://www.loc.gov/z3950/> (Дата обращения 03.05.2021). Загл. с экр. Яз. англ.
- 15 Система ИРБИС64+ [Электронный ресурс]. — URL: <https://tinyurl.com/ye171rw2> (Дата обращения 03.05.2021). Загл. с экр. Яз. рус.
- 16 *Бродовский, А. С.* Перспективы развития системы ирбис: Новый продукт ирбис64+ / А. С. Бродовский // *НАУЧНЫЕ И ТЕХНИЧЕСКИЕ БИБЛИОТЕКИ*. — 2017. — Vol. 11. — Pp. 65–74.
- 17 РОССИЙСКИЙ КОММУНИКАТИВНЫЙ ФОРМАТ [Электронный ресурс]. — URL: <http://www.rusmarc.ru/> (Дата обращения 03.05.2021). Загл. с экр. Яз. рус.
- 18 Система дистанционного обучения IpsilonUni [Электронный ресурс]. — URL: <https://ipsilon.sgu.ru/> (Дата обращения 04.05.2021). Загл. с экр. Яз. рус.
- 19 Slim - A Fast, Lightweight Template Engine for Ruby [Электронный ресурс]. — URL: <http://slim-lang.com/> (Дата обращения 05.05.2021). Загл. с экр. Яз. англ.
- 20 HTML Standart [Электронный ресурс]. — URL: <https://html.spec.whatwg.org/> (Дата обращения 05.05.2021). Загл. с экр. Яз. англ.
- 21 Ruby on Rails [Электронный ресурс]. — URL: <https://rubyonrails.org/> (Дата обращения 05.05.2021). Загл. с экр. Яз. англ.
- 22 What Is Linux? [Электронный ресурс]. — URL: <https://www.linux.com/what-is-linux/> (Дата обращения 08.05.2021). Загл. с экр. Яз. англ.

- 23 ManagedIrbis [Электронный ресурс].— URL: <https://github.com/amironov73/ManagedIrbis> (Дата обращения 15.05.2021). Загл. с экр. Яз. англ.
- 24 ФОРМАТ ПРЕДСТАВЛЕНИЯ ДАННЫХ В ЭЛЕКТРОННОМ КАТАЛОГЕ ИРБИС64+ [Электронный ресурс].— URL: <http://sntnarciss.ru/irbis/spravka/irbis64.html?prm100600000.htm> (Дата обращения 20.05.2021). Загл. с экр. Яз. рус.
- 25 IIS Express Overview [Электронный ресурс].— URL: <https://docs.microsoft.com/en-us/iis/extensions/introduction-to-iis-express> (Дата обращения 21.05.2021). Загл. с экр. Яз. англ.
- 26 Visual Studio [Электронный ресурс].— URL: <https://visualstudio.microsoft.com/ru/> (Дата обращения 21.05.2021). Загл. с экр. Яз. рус.