# **1 ОБЗОР ЛИТЕРАТУРЫ**

На этапе начала проектирования программного модуля для системы были учтены требования того, что она должна быть запущена и функционировать под управлением операционной системы Windows Server 2008.

Для реализации поставленных требований была выбрана платформа разработки Microsoft .NET и язык программирования C#, как наиболее современный и распространённый на данной платформе.

К настоящему моменту наиболее популярными объектно-ориентированными языками программирования являются C#, Java, C++ и Objective-C. Учитывая преимущества и недостатки каждого из них, был сделан выбор в пользу языка C# и платформы Microsoft .NET Framework.

Microsoft .NET Framework – программная технология, предназначенная для создания как обычных программ, так и веб-приложений. Фактически представляет собой операционную систему внутри операционной системы, высокопроизводительную, основанную на стандартах, многоязыковую среду, которая позволяет интегрировать существующие приложения с приложениями и сервисами следующего поколения, а также решать задачи развертывания и использования интернет-приложений. .NET Framework состоит из трех основных частей - общеязыковой среды выполнения (Common Language Runtime), иерархического множества унифицированных библиотек классов и компонентную версию ASP, называемую ASP.NET.. Основой платформы является [виртуальная машина](http://ru.wikipedia.org/wiki/%D0%92%D0%B8%D1%80%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D0%B0) [Common Language Runtime (CLR)](http://ru.wikipedia.org/wiki/Common_Language_Runtime), способная выполнять как обычные настольные программы, так и веб-приложения. Одной из основных идей Microsoft .NET является совместимость различных служб, написанных на разных языках. Например, служба, написанная на C++ для Microsoft .NET, может обратиться к методу класса из библиотеки, написанной на Delphi; на C# можно написать класс, наследованный от класса, написанного на Visual Basic .NET, а исключение, созданное методом, написанным на C#, может быть перехвачено и обработано в Delphi. Каждая библиотека (сборка) в .NET имеет сведения о своей версии, что позволяет устранить возможные конфликты между разными версиями сборок. Так же как и технология Java, среда разработки .NET создаёт байт-код, предназначенный для исполнения виртуальной машиной. Входной язык этой машины в .NET называется MSIL (Microsoft Intermediate Language), или CIL (Common Intermediate Language, более поздний вариант), или просто IL. Применение байт-кода позволяет получить кроссплатформенность на уровне скомпилированного проекта (в терминах .NET – сборка), а не только на уровне исходного текста, как, например, в С. Перед запуском сборки в среде исполнения CLR, байт-код преобразуется встроенным в среду JIT-компилятором («just in time» ­– компиляция на лету, англ.) в машинные коды целевого процессора. Также существует возможность скомпилировать сборку в родной код для выбранной платформы.

Описание базовых принципов функционирования платформы .NET, системы типов .NET и различных инструментальных средств разработки, используемых при создании приложений, базовые возможности языка программирования С#, включая новые синтаксические конструкции, появившиеся с выходом .NET 4.0, а также синтаксис и семантика языка CIL подробно описаны в [1]. Также там рассматривается формат сборок .NET, библиотеки базовых классов .NET файловый ввод-вывод, возможности удаленного доступа, конструкция приложений Windows Forms, доступ к базам данных с помощью ADO.NET. Там описаны основные объекты модели ADO.NET и рассказано как сохранять, искать, фильтровать и сортировать данные с использованием этих объектов. Книга была очень полезна, так как организована как справочное пособие и позволяет быстро найти ответы на интересующие вопросы. Здесь рассматриваются как подсоединенные, так и отсоединенные объекты, в том числе DataAdapter.DataSet, Connection, Command, Transaction, DataReader и др. Отдельный раздел книги посвящен созданию эффективных Windows- и Web-приложений для доступа к БД с использованием ADO.NET.

Одним из лучших практических руководств по программированию можно назвать «Совершенный код». Оттуда почерпнуты эффективные методики и принципы программирования, которые были использованы при разработке.

Более детальная информация с рекомендациями по использованию конкретных классов приведена в [2]. В ней рассматривается подробное описание внутреннего устройства и функционирования общеязыковой исполняющей среды (CLR) Microsoft .NET Framework версии 2.0, система типов .NET Framework и разъяснены способы управления ими. Представлены концепции программирования с широким использованием библиотеки FCI, относящиеся ко всем языкам, ориентированным на работу с .NET Framework. Особое внимание уделено обобщениям, управлению асинхронными операциями и синхронизации потоков.