

ЧАСТЬ I

Спринт первый: Приложение Alpine Ski House

Глава 1. Что к чему.....	25
Глава 2. Факторы влияния	40
Глава 3. Модели, представления и контроллеры.....	52
Глава 4. Структура проекта.....	62
Глава 5. Сборка кода	72
Глава 6. Развертывание	88

Пора познакомиться с художественной частью материала, включая вымышленных персонажей, создающих приложение Alpine Ski House.

Даже самые рьяные лыжники признали: сезон подходил к концу. Это был не лучший сезон на их памяти, но он не был и худшим. В общем-то сезон был во всех отношениях неприметным. Конечно, в нем были кое-какие запоминающиеся моменты: из-за перебоев с электричеством в конце февраля пришлось пустить в ход план действий в аварийных ситуациях, который часто отработывался, но еще никогда не применялся. На местных каналах новостей были сообщения о детях, которые застревали в кабинках на несколько часов, но благодаря мягкой погоде им в действительности ничего не угрожало. Акций с раздачей бесплатных абонементов было достаточно, чтобы лыжники и сноубордисты приезжали снова и снова.

Весна — время, когда происходила перегруппировка постоянного персонала, а сезонные работники перебирались в теплые края. Среди постоянного персонала ходили слухи, что по крайней мере половину сезонных работников, как только

они сходили со склона, перехватывала иммиграционная служба и отправляла обратно в Австралию. Даниэль не могла представить, почему молодые сезонные работники должны были сопротивляться отправке в Австралию. Можно было смело утверждать, что в Австралии им намного интереснее, чем в сонном горном городке, который пробуждался от спячки только зимой.

Строить планы на следующий год было еще рано, и Даниэль надеялась, что у нее образуется месяц-другой свободного времени, прежде чем все завертится по новой. Она уже почти 10 лет была единственным разработчиком в Alpine Ski House, и каждый год происходило одно и то же. Большая часть ее рабочего времени уходила на поддержание работы унаследованных систем и настройку всяких мелочей, которые могли понадобиться в следующем году. Работа была не самая интересная, но в зимние месяцы в хорошие дни всем работникам удавалось провести пару часов на склоне, и это ей очень нравилось.

Даниэль открыла дверь невысокого домика, где размещалась компания Alpine Ski House, и замерла от удивления — в помещении шла какая-то бурная деятельность. Люди, которых ждали здесь не раньше чем через два часа, уже были на месте и заполняли небольшими группами все открытое пространство офиса. Озадаченная Даниэль бросила сумку, налила себе кофе и стала искать, к кому бы подойти. Похоже, весь IT-персонал собрался возле Тима — дородного руководителя IT-отдела и ее начальника. Даниэль подошла к нему.

«Даниэль! Слышала последние новости? Нас ждет интересное время, вот что я скажу», — громогласно заявил Тим.

«А что за новости?» — спросила Даниэль.

«Ты что, не в курсе? — спросил Арджун. — Наша компания только что завершила сделку по покупке курортов Thunder Valley и Mount Ballyhoo. Происходит слияние, и мы все потеряем работу!»

Два других лыжных курорта располагались всего в нескольких милях. Thunder Valley — маленькое заведение всего с тремя подъемниками, которое, тем не менее, имело своих сторонников. Этот курорт был популярен у местного населения, которое хотело отдохнуть от толп туристов в зимние месяцы. Он разительно отличался от Mount Ballyhoo — громадного курорта, охватывавшего целых три горы, с бесчисленными подъемниками и гостиницами, в которых могли бы разместиться жители двух таких городков. По выходным на склоне играли оркестры, а в толпе нередко можно было встретить таких знаменитостей, как Скотт Гатри (Scott Guthrie) и Джон Скит (John Skeet).

«Послушай, Арджун, — сказал Тим, — никто ничего не говорил об увольнении, сокращениях или о чем-то подобном. В такие времена нагрузка на IT всегда возрастает, и только лишь потому, что руководство хочет интегрировать все системы должным образом. Нужно просто подождать и понять, куда ветер дует».

Даниэль растерянно опустила на стул. Ей совершенно не хотелось искать другую работу. Сколько вакансий для программиста найдется в таком городке с сезонным наплывом туристов? «Глупости, — сказала она себе, — пока нет смысла планировать переезд в большой город при такой неопределенности. В следующую пару недель ситуация утрясется».

Как оказалось, пару недель ждать не пришлось.

Как только закончился обед (салат из одуванчиков и грецких орехов с картофельными чипсами), Тим подошел к ее рабочему месту.

«Мы собираемся в конференц-зале. Похоже, программисты из Thunder и Ballyhoo уже там».

Допив козье молоко, Даниэль схватила ручку и блокнот и поспешила в конференц-зал. Все это было только для виду; в последние годы она не делала заметки на собраниях. Было проще лично встретиться с людьми в маленьком офисе, чем заранее планировать встречи в блокноте. Впрочем, если на горизонте маячат увольнения, лучше сразу произвести хорошее впечатление.

Большой конференц-зал использовался редко, потому что обычно просто не хватало людей, чтобы его заполнить. Но сегодня он был если и не набит до отказа, то по крайней мере не казался пустым. У одного конца стола сидели пятеро молодых парней хипстерской наружности с коктейлями-смужи самого экзотического вида.

В явном отдалении от хипстеров сидела группа людей, которых можно было безошибочно отнести к начальству. Они были загорелыми и беззаботными, и, казалось, только что вышли с поля для гольфа.

Тим подождал, пока шум стихнет, и обратился к аудитории: «У меня хорошие новости. Мы все — команда, и идти вперед нам предстоит вместе. Так что, если вы находитесь в этой комнате, значит, можно не беспокоиться; с вашей работой все в порядке. Наверняка у всех вас есть вопросы, если кто-то захочет поговорить — подойдите ко мне после собрания».

«Руководство предложило мне увеличить штат программистов после слияния, потому что нам предстоит работа над новыми интересными проектами. За несколько следующих лет мы обновим все наши программные системы, которые сейчас находятся в эксплуатации. Руководство понимает, что это серьезная работа, кое-кто из них даже читал один умный журнал и теперь знает о методологии Agile и микросервисах. Я распорядился, чтобы в будущем все экземпляры журнала сжигали до того, как они попадут в руки начальства, но сейчас нам от этого никуда не уйти».

У Тима всегда были непростые отношения с новыми «великими» идеями руководства. Он работал аварийным тормозом для их безумных затей. Тим продолжил: «Первое, чего от нас хотят, — чтобы люди могли покупать билеты на подъемник

через интернет. Мне напомнили, что на дворе 2016 год и что эта функциональность должна была быть реализована давным-давно, тем более что на всех остальных курортах в мире она уже есть». Тима явно раздражали эти обобщения; вероятно, когда он получал распоряжения, дискуссия была жаркой.

«Начальство хочет увидеть прототип через месяц. Я думаю, что смогу выторговать еще неделю, если мы покажем, что работа не стоит на месте».

«Через месяц!» — подумала Даниэль. Месяц обычно требовался ей только для того, чтобы разобраться в сути проблемы. Она взглянула на хипстеров, надеясь, что они сидят с такими же кислыми лицами, как у нее, однако любители здорового питания кивали с довольным видом.

Тим, кажется, готовился наконец-то закончить свою речь. «Послушайте, ребята, нам это нужно, чтобы заработать репутацию у нового руководства. Я сделаю все, чтобы работа шла гладко. Используйте любые технологии, которые, на ваш взгляд, лучше подойдут; любые инструменты, которые вам понадобятся. Верю, что вы справитесь».

1

Что к чему

Завершив свою речь, Тим с командой гольфистов покинул комнату, и Даниэль осталась наедине с пятью программистами-хипстерами. Руководить новой группой разработки, конечно же, никого не назначили. Скорее всего, Тим тоже немного почитал злополучный журнал для IT-директоров и решил, что лучшая группа — это группа, способная к самоорганизации. Даниэль подумала, что как-нибудь стоит сунуть курьеру двадцатку, чтобы в будущем такие журналы «случайно терялись» — наверняка в следующем году она больше сэкономит на средствах от головной боли.

Хипстеры представились: Адриан, Честер, Кэндис и два Марка (Даниэль окрестила их Марк-1 и Марк-2). Все они пришли с курорта Mount Ballyhoo, руководство которого за несколько недель до этого решило отказаться от внешних разработчиков и нанять полную команду программистов в штат. Хипстеры работали вместе в стартапе в Кремниевой долине, и в Mount Ballyhoo наняли сразу всех пятерых (эта идея показалась Даниэль довольно удачной). Программисты даже не успели запустить ни один проект в Ballyhoo к тому моменту, как их выкупили. Выглядели они вполне дружелюбными и немного взволнованными.

«Мне понравилось, что руководство дает нам самим выбирать технологии, — отметил Марк-2. — Это означает, что они нам доверяют и не хотят решать за нас».

«Выбор фреймворка всегда был моей любимой частью проекта», — заметила Кэндис.

«Сейчас есть миллион отличных вариантов: Phoenix, Revel, Express, даже Meteor. А ты что предпочитаешь, Даниэль?» — спросил Адриан.

Phoenix? Express? Похоже, что авторы называли свои веб-фреймворки в честь любимых китайских ресторанов. Даниэль впервые слышала все эти названия — она была разработчиком ASP.NET до мозга костей. Когда-то она писала приложения WinForms, а весь последний год проработала с ASP.NET MVC. Отбросив сомнения, она ответила: «Обычно я использую ASP.NET».

«О, да! — воскликнул Марк-2. Даниэль показалось, что говорить на громкости ниже 90 децибел он вообще не способен. — Одна группа в Microsoft делает со-

вершенно невероятные штуки с ASP.NET Core. Ты бы видела их производительность! Полный улет!»

Даниэль не была уверена в том, что именно «полный улет» означает в этом контексте. Вроде бы ничего плохого, поэтому она кивнула в знак согласия.

«Хмм, — сказала Кэндис. — Я не ожидала, что речь пойдет об ASP.NET».

«Да ладно, это новая и крутая технология, — сказал Марк-1. — Даниэль, у тебя ведь есть опыт работы с ASP.NET. Можешь в двух словах рассказать, что к чему?»

Веб-разработка уже давно перестала быть экзотическим новшеством. На заре интернета веб-страницы были статическими, а вся навигация обеспечивалась посредством гиперссылок. Недостатки такого подхода быстро стали очевидными. Разработчику было нелегко выдержать единую тему оформления на всех страницах, а просмотр одинаковых веб-страниц вызывал у пользователей скуку.

Веб-серверы начали поддерживать технологии, благодаря которым вид веб-страницы мог изменяться в зависимости от входных данных. Одни технологии приходили и уходили (например, SSI — Server Side Includes), другие — такие, как CGI (Common Gateway Interface), — в той или иной форме продолжали существование. Название Active Server Pages, или ASP, используется уже около двадцати лет, в то время как сама технология претерпела радикальные изменения. Сейчас мы стоим перед очередным кардинальным преобразованием технологии, скрывающейся за термином ASP. Прежде чем мы погрузимся в ее изучение, давайте сначала разберемся, какой путь она прошла.

Active Server Pages

В 1996 году третья версия сервера IIS (Internet Information Services) была выпущена с поддержкой ASP первой версии. Технология ASP строится на базе технологии Active Scripting. При этом применение Active Scripting не ограничивается ASP, так как эта технология также стала частью Internet Explorer и Windows Script Host.

Технология ASP позволяет встраивать в веб-страницы сценарии, написанные на другом языке, и исполнять их. Теоретически, поддержка большинства языков обеспечивается посредством интеграции Active Scripting с технологией COM (Component Object Model). В те времена основная конкуренция в этой области шла между JScript и VBScript — двумя языками, которые пользовались хорошей поддержкой со стороны Microsoft. Впрочем, в середине 1990-х другие языки — такие, как Perl, — тоже были популярными. Язык Perl до определенной степени обеспечивал кроссплатформенную совместимость, так как он поддерживался в Linux и мог работать через CGI-шлюзы Apache. Perl был одним из самых популярных

языков для создания интерактивных веб-приложений в ранних версиях веб-сервера Apache, являясь предшественником NCSA HTTPd.

Рассмотрим пример файла, в котором в качестве языка сценариев используется VBScript. Давайте удостоверимся, что при выводе используются современные теги на базе HTML. На самом деле, ничто не мешает использовать ASP для построения современных приложений.

```
<%@ Language= "VBScript" %>

<html>
  <head>
    <title>Example 1</title>
  </head>
  <body>
    <header>
      <h1>Welcome to my Home Page</h1>
    </header>
    <section id="main">
      <%
        dim strDate
        dim strTime

        'Получение даты и времени.
        strDate = Date()
        strTime = Time()

        ' Вывод разных приветствий в зависимости от времени суток
        If "AM" = Right(strTime, 2) Then
          Response.Write "<span>Good Morning!</span>"
        Else
          Response.Write "<span>Good Afternoon!</span>"
        End If
      %>
      Today's date is <%=strDate %> and the time <%=strTime%>

    </section>
  </body>
</html>
```

Этот пример демонстрирует различные подходы к передаче данных в выходной поток. Первый подход — напрямую обратиться к ответному потоку посредством `Response.Write`, как это сделано для вывода приветствия. При втором подходе используется директива `<%= %>` — сокращенная форма записи в ответный поток. Этот способ очень похож на те механизмы, которые встречаются в других языках сценариев, например в PHP. Компилятор игнорирует все теги HTML, кроме специальных тегов, содержащих директивы. Такие теги распознаются по последовательности символов `<%`. Директива `<%@` предоставляет компилятору ASP информацию, необходимую для обработки страницы. В приведенном примере данная

директива задает основной язык для оставшейся части страницы. Простая директива `<%` обозначает простой блок программного кода.

В отличие от PHP, технология Active Server Pages позволяет организовать совместное использование библиотек посредством COM. Так страницы могут обращаться к низкоуровневым компонентам, работающим с высокой эффективностью, или пользоваться низкоуровневой функциональностью. Кроме того, эта возможность позволяет создавать библиотеки компонентов, которые улучшают структуру приложения и способствуют повторному использованию кода.

В конце 1990-х годов компания Microsoft приступила к созданию собственной платформы разработки следующего поколения. В то время она называлась Next Generation Windows Services, или NGWS. К концу 2000-х она преобразовалась в то, что мы сейчас знаем как платформу .NET. Эта платформа включала переработанную версию популярного языка Visual Basic. При переработке в язык была включена поддержка объектно-ориентированного программирования — весьма радикальное изменение. Во многих отношениях новая версия, которая стала называться VB.NET, стала совершенно новым языком.

В .NET была включена хорошо спроектированная библиотека базовых классов, которая предоставляла большой объем стандартной функциональности. На разработку BCL (Base Class Library) значительно повлияла аналогичная библиотека, заложенная в основу Java.

Также в .NET появился совершенно новый язык, который назывался C#; он также разрабатывался под сильным влиянием Java. Язык C# предоставлял C-образный синтаксис, который лучше подходил программистам с опытом работы на C и Java. Компания Microsoft интенсивно продвигала C#, и в результате этот язык стал самым популярным языком платформы .NET.

Вместе с новыми языками C# и Visual Basic .NET была представлена новая версия ASP, которая была названа ASP.NET.

ASP.NET

Уже в конце 1990-х годов стало очевидно, что Всемирная паутина — нечто большее, чем преходящее увлечение. Но, к несчастью, у компании Microsoft возникла одна проблема. Компания потратила годы на создание инструментов, позволявших проектировать приложения для настольных систем в визуальном редакторе. Разработчики привыкли придерживаться именно такого подхода в построении приложений, а компаниям совершенно не хотелось оплачивать переподготовку всего своего персонала. Взаимодействие с элементами управления обеспечивалось по модели обработки событий. Кнопке на форме назначался обработчик события, который выполнял некоторое скрытое от глаз действие, после чего в пользовательском интерфейсе (UI, User Interface) происходили изменения. Со-

общество разработчиков привыкло к этому методу разработки, и, что еще важнее, он оказался чрезвычайно эффективным. До сих пор не существует практически ни одного инструмента, способного сравниться по эффективности с принципом генерирования кода WYSIWYG (What You See Is What You Get — «Что видишь, то и получаешь»), применяемым в Visual Basic, а позднее и в WinForms. Возможно, визуальный интерфейс получался не самым впечатляющим, но если вашему продукту не требовалось ничего, кроме серых прямоугольников на сером фоне, разработка в Visual Basic 6 или WinForms давала далеко не худший результат.

Технология ASP.NET Web Forms стала попыткой перенести эту эффективность в веб-разработку. В ней была реализована аналогичная функциональность перетаскивания мышью для размещения элементов управления по сетке и взаимодействия с ними на уровне откомпилированного кода на стороне сервера.

К сожалению, модель обратной передачи (postback), которая использовалась для отправки взаимодействий на сервер, игнорировала фундаментальные различия между веб-приложениями и настольными приложениями; каждый вызов для обратной передачи данных и выполнения действия на сервере требовал полного цикла HTTP «запрос — ответ» по интернету. Несмотря на этот недостаток, технология ASP.NET Web Forms стала исключительно успешным продуктом, на базе которого работают бесчисленные сайты — как внутренние сайты компаний, так и общедоступные сайты в интернете.

Страницы Web Forms обычно состоят из двух частей: представления (view) и отделенного кода (code behind). Представление создается в специальном синтаксисе, объединяющем HTML с Visual Basic или C#. Такие файлы имеют расширение .aspx и обрабатываются компилятором ASP. По поводу того, какая информация должна содержаться в таких файлах, нет единого мнения. У разработчика имеется определенная свобода выбора, поэтому одни люди размещают в файле всю логику, а другие ограничиваются только логикой, относящейся к представлению. Известны успешные проекты, в которых использовались оба подхода. Отделенный код компилируется в DLL, это означает, что вносить изменения «на лету» не получится. Файл .aspx компилируется «на лету», а это означает, что изменения в серверную часть могут быть внесены посредством обычного текстового редактора (хотя поступать так крайне не рекомендуется).

Каждая страница реализуется в виде формы HTML. Это позволяет Web Forms легко отслеживать изменения состояния и реагировать на них обратной передачей формы. Если вы когда-либо видели сайт, который перезагружает всю страницу целиком при отведении указателя мыши от поля ввода данных, скорее всего, вы имели дело с приложением Web Forms, которое выполняло обратную передачу состояния страницы на сервер для принятия решений о том, какие элементы страницы подлежат изменению на основании введенных данных. Модель взаимодействия, мягко говоря, не блестящая, но она может быть улучшена при помощи технологий AJAX и Update Panels, которые ограничиваются обратной передачей только отдельных частей страницы.