

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Стрельчик С.А.

Сечко Г. В. – канд. техн. наук, доцент

Рассматриваются элементы функционального программирования в .NET FRAMEWORK. Работа является продолжением аналогичных исследований, изложенных в работах студентов ИИТ БГУИР [1-3] и работах других исследователей [4].

На протяжении всей короткой истории развития компьютерных технологий от магистральной линии развития языков программирования время от времени отвечались различные практические и теоретические направления. Языки 4-го поколения, появившиеся в 90-ых годах, представляют собой практическое ответвление, а функциональное программирование, появившееся с языком LISP в конце 50-ых, – это пример теоретической ветви. Со временем многие ответвления вновь присоединялись к основному ядру, и именно это сейчас и происходит с функциональным программированием.

Чтобы решить задачу, разработчики объектно-ориентированных приложений проектируют иерархии классов, добиваются правильной инкапсуляции и мыслят в терминах контрактов между классами. Для них первостепенное значение имеет обеспечение правильного поведения и состояния типов объектов, а для достижения этого используются такие возможности языка, как классы, интерфейсы, наследование и полиморфизм.

В отличие от этого, в функциональном программировании применяется подход к вычислительным проблемам как к определению чисто функциональных преобразований коллекции данных. В функциональном программировании приходится отказываться от применения состояний и изменяющихся данных, а вместо этого сосредотачиваться на применении функции.

Принципы функционального программирования формулировались специально для поддержки чисто функционального подхода к решению проблем. При императивном подходе разработчик пишет код, подробно определяющий шаги, которые должен выполнить компьютер для достижения цели. Такое программирование иногда называют алгоритмическим. В отличие от него, функциональный подход сводится к составлению решения задачи в виде набора функций, которые должны быть выполнены. Разработчик подробно определяет вход каждой функции и возвращаемые ею результаты. Традиционное объектно-ориентированное программирование является таковым, что большинство разработчиков привыкает писать код в императивном (процедурном) стиле. Переходя к разработке в чисто функциональном стиле, они должны изменить свое мышление и подход к разработке.

Функциональные языки появляются не только для виртуальной Java-машины, где наиболее интересны два новых языка – Scala и Clojure, но и для платформы .NET, где появился первоклассный язык F#. Причина тому, что все платформы начали приобретать возможности функционального программирования заключается в том, что со временем среда исполнения стала способной выполнять все большее количество рутинной работы, а разработчики получили возможность передавать ей все больше контроля над выполнением подобных задач.

Программы на F# обычно получаются лаконичнее. При этом сокращается не только количество символов, но и количество мест, где компилятор требует указать тип переменной, аргумента или возвращаемых значений. Значит, и обслуживать приходится меньший объем кода. По производительности F# сравним с C#, однако, если сравнивать с языками, которые настолько же лаконичны, как F#, в частности, с динамическими языками и языками сценариев, он имеет гораздо лучшие показатели. И также как во многих динамических языках, в F# имеются средства, позволяющие проверить данные: написать фрагмент кода и запустить его в интерактивном режиме.

Список использованных источников:

1. Аксёнов, В. И. PYTHON как язык, поддерживающий несколько парадигм программирования // 49-я науч. конф. аспирантов, магистрантов и студентов БГУИР по направлению 8: Информационные системы и технологии: тез. докл. (Минск, 4 мая 2013 года). – Мн.: БГУИР, 2013. – 91 с. с ил. – С. 59-60.
2. Селивошко, В. И., Шинкевич А. А. Функциональное программирование в JVM // 49-я науч. конф. аспирантов, магистрантов и студентов БГУИР по направлению 8: Информационные системы и технологии: тез. докл. (Минск, 4 мая 2013 года). – Мн.: БГУИР, 2013. – 91 с. с ил. – С. 81-83.
3. Корablёв, Д. А. Элементы функционального программирования в языке RUBY // 50-я науч. конф. аспирантов, магистрантов и студентов БГУИР по направлению 8: Информационные системы и технологии: тез. докл. (Минск, 29 марта 2014 года). – Мн.: БГУИР, 2014. – 78 с. с ил. – С. 23.
3. Функциональное мышление: Часть 1. Разработка программ в функциональном стиле [Электронный ресурс] – Режим доступа: <http://www.ibm.com/developerworks/ru/library/j-ft1/index.html>. – Дата доступа: 15.03.2015.

ПРОГРАММНОЕ СРЕДСТВО УПРАВЛЕНИЯ ВЗАИМОДЕЙСТВИЕМ С КЛИЕНТАМИ ЛОГИСТИЧЕСКОЙ КОМПАНИИ

Институт информационных технологий БГУИР,

Обоснована необходимость разработки программного средства для управления взаимодействием с клиентами логистической компании. Рассмотрены основные модули и функциональные возможности разрабатываемой системы.

Логистика – это система наук управления продвижением материалопотока, то есть управление материалами и управление распределением. Целью логистики является обеспечение получения (доставки) продукции потребителю в нужное время и место при минимально возможных совокупных затратах трудовых, материальных, финансовых ресурсов [1]. Количество перемещаемых грузов по всему миру чрезвычайно велико. Работа логистических компаний подразумевает ежедневный прием значительного числа заявок, их обработку и анализ, хранение требуемой информации, а также, при необходимости, использование ее для построения различного рода отчетов. В связи с этим в последнее время ощущается все больший интерес к автоматизации со стороны активно развивающихся компаний, владеющих достаточным количеством средств, желающих минимизировать потери рабочего времени персонала и поднять уровень сервиса. Разрабатываемое программное решение, позволит оптимизировать блок задач верхнего уровня для менеджеров по работе с клиентами и других сотрудников компании. При разработке будет использоваться модульный подход, поэтому код приложения будет выделен в два модуля: «front-office» и «back-office».

Система привилегий является самым оптимальным решением по разграничению прав доступа, поэтому были выделены следующие системные роли пользователей:

- гость – не имеет доступа к редактированию и удалению сущностей, но может просматривать статические информационные страницы системы (например, «Контакты»), то есть доступ к «front-office»;
- клиент – не имеет доступа к редактированию и удалению сущностей, но может просматривать свои заказы;
- менеджер – имеет права на редактирование данных в «back-office», относящихся к заказам компаний;
- администратор – имеет доступ ко всем ресурсам, к тому же возможность управления настройками системы, а также будет предоставлена возможность создавать новые группы пользователей с произвольными названиями и правами доступа.

Доступ к «back-office» будет доступен только авторизованным пользователям, имеющим соответствующие права (по умолчанию менеджеры и администраторы). Создание новых пользователей будут осуществлять пользователи с правами администратора. Открытой регистрации пользователей не будет. Это будет сделано из-за соображений безопасности.

Главный алгоритм системы следующий: пользователь со страницы «Отправить заявку» отправляет все требуемые данные для доставки груза. К таким данным будут относиться «Место отправки», «Место доставки», «Дата отправки», «Дата доставки», дополнительные флаги, например «Срочный заказ» и так далее. Менеджер на странице со списком заявок осуществляет обработку полученных заявок. Под обработкой можно понимать связь с отправителем заявки по телефону, электронной почте, для уточнения деталей заявки, прокладка маршрута доставки и так далее. Корректные заявки отправляются менеджерами в заказы на исполнение, а для отправителя администратором создается новый пользователь в системе, если таковой не был создан ранее. В личном кабинете клиенты смогут отслеживать статус заказа, процесс и прогресс доставки. Также будет предусмотрена возможность выбора типа транспорта, управление складами. Для страниц со списками будет реализован функционал фильтров и полнотекстового поиска. Модуль «front-office» будет представлен статическими информационными страницами (например, «Контакты»), а также страницей, содержащей форму обратной связи. Для анализа активностей в системе предусмотрено ведение журнала действий, производимых пользователями. Администраторы смогут в любой момент просмотреть этот журнал на соответствующей странице в «back-office». Главным требованием к системе является доступ онлайн, поэтому программное средство разрабатывается в виде web-приложения. В качестве серверного языка программирования будет использоваться PHP (выбор сделан в пользу его кроссплатформенности: запуск на всех популярных серверах под управлением самых различных операционных систем) с использованием фреймворка Symfony 2, а клиентского – JavaScript. В качестве системы управления базами данных была выбрана MySQL.

Таким образом, разрабатываемое программное средство позволит менеджерам и другим сотрудникам логистической компании иметь в своем распоряжении удобный инструмент для управления взаимодействием с клиентами компании.

Список использованных источников:

1. Большая энциклопедия нефти и газа. [Электронный ресурс].– Режим доступа; <http://www.ngpedia.ru/id138976p4.html>. – Дата доступа 26.03.2015

АНАЛИЗ ПРЕИМУЩЕСТВ ИСПОЛЬЗОВАНИЯ ПЛАТФОРМЫ ARDUINO

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь