



Рисунок 2 – Функциональная схема автоматизированной системы подключения абонентов к сети

Таким образом, в данном проекте была разработана автоматизированная система подключения абонентов к сети передачи данных, которая позволила удаленно управлять подключением и отключением абонентов, отслеживать на порту текущее состояние каждого абонента, осуществлять блокировку и разблокировку абонентов при необходимости и многое другое. В результате повысится производительность и эффективность труда работников, улучшится качество услуг и оперативность обслуживания пользователей. Также сократится время на подключение абонента к услуге ADSL, и тем самым увеличится количество подключаемых абонентов. Более эффективно будет использоваться транспорт, предоставленный для обслуживания абонентов, в связи с автоматическим планированием выездов к абонентам.

В программном комплексе также разработан механизм для учёта оборудования, переданного пользователям ADSL, что позволяет контролировать перемещение комплектов оборудования и сохранения их целостности.

Функции администрирования существующего оборудования максимально упрощены и автоматизированы.

Внедрение системы позволит существенно сократить время обслуживания пользователей, значительно повысить уровень их обслуживания.

Список использованных источников:

1. Гайдамакин, Н.А. Автоматизированные информационные системы, базы и банки данных./ Н.А.Гайдамакин. - М.: Наука, 2002- 394 с.
2. Липаев, В.В. Надежность программного обеспечения АСУ. / В.В. Липаев.– М.: Энергоиздат, 1999- 326 с.
3. Мацяшек Л.А. Анализ требований и проектирование систем./ Л.А. Мацяшек. - М.: Наука, 2002- 352 с.
4. Титоренко, Г. А. Информационные технологии управления./ Г. А. Титоренко. - М.: ЮНИТИ, 2002- 344 с.
5. Шураков В. В. Автоматизированное рабочее место для статической обработки данных./ В. В. Шураков. - М.: Наука, 1990 - 246 с.

ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ И .NET FRAMEWORK

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. Кораблёв, 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,