

Министерство образования Республики Беларусь
Учреждение образования «Барановичский государственный университет»
Студенческое научное общество БарГУ

СОДРУЖЕСТВО НАУК. БАРАНОВИЧИ-2016

Материалы XII Международной
научно-практической конференции
молодых исследователей

(Барановичи, 19—20 мая 2016 года)

В трёх частях

Часть 2

Барановичи
БарГУ
2016

В части 2 сборника материалов XII Международной научно-практической конференции молодых исследователей «Содружество наук. Барановичи-2016» представлены результаты исследований в области физики и математики, а также рассмотрены актуальные проблемы в области информационных систем и технологий в образовании, науке и технике. Особое внимание уделено современным тенденциям в технологиях и материалах машиностроительного и сельскохозяйственного производств, а также экономическим аспектам развития предприятия, региона.

Сборник адресован научным работникам, аспирантам, магистрантам и студентам инженерных и экономических специальностей учреждений высшего образования.

Редакционная коллегия:

А. В. Никишова (гл. ред.), Ю. Е. Горбач, В. Н. Кременевская (отв. секретари), Е. Н. Кирюхова,
О. И. Наранович, А. К. Гавриленя, М. В. Нерода, В. Н. Познякевич, Г. Я. Житкевич

Рецензент

кандидат технических наук, заведующий лабораторией механофизики гетерогенных систем
Государственного научного учреждения «Физико-технический институт
Национальной академии наук» А. М. Милюкова

Научное издание

СОДРУЖЕСТВО НАУК.
БАРАНОВИЧИ-2016

Материалы XII Международной
научно-практической конференции
молодых исследователей

(Барановичи, 19—20 мая 2016 года)

На русском, белорусском, английском языках

В трёх частях

Часть 2

Ответственный за выпуск Е. Г. Хохол
Технический редактор А. Ю. Сидоренко
Компьютерная вёрстка С. М. Глушак
Корректор Н. Н. Колодко

Подписано в печать 04.10.2016. Формат 60 × 84 ¹/₈. Бумага ксероксная.

Отпечатано на копировально-множительной технике. Усл. печ. л. 28,00. Уч.-изд. л. 25,10. Тираж 9 экз. Заказ 681.

Учреждение образования «Барановичский государственный университет».
Свидетельство о государственной регистрации издателя № 1/424 от 09.09.2016.
Ул. Войкова, 21, 225404 г. Барановичи. Тел. 8 (0163) 45 46 28, e-mail: rio@barsu.by .

широко известных методов выделения частоты основного тона (например, корреляционный метод, кепстральный метод, метод Голда—Рабинера) [3].

Если первая группа признаков отражает статические свойства речеобразующего тракта, то вторая группа призвана полностью описать его поведение во времени, т. е. артикуляционную динамику речи. Согласно существующему предположению, исходным и основным этапом в организации процесса речеобразования является управляемая центральной нервной системой человека программа комплекса артикуляционных движений, соответствующая тому сообщению, передача которого планируется в данный момент времени.

Для расчёта параметров, описывающих артикуляционную динамику речи, могут быть использованы методы спектрально-временного анализа данных, описанные выше. Однако необходимо отметить такую особенность расчёта просодических параметров, как их жёсткая связь с лексическим и синтаксическим контекстом исследуемой фразы. Это требует комплексного применения как средств лингвистического анализа, так и параметрических методов обработки, что явно определяет сложность анализа данных характеристик. При этом основной задачей является установление прямой связи между деятельностью речеобразующего аппарата (динамикой его артикуляционных движений) и характеристиками спектральной картины потока речи [4].

Большинство разработанных на сегодня систем идентификации личности по голосу построены на основе однократной проверки соответствия требуемой ключевой фразы и произнесённой в первоначальный момент доступа к вычислительной системе. Данные системы поддерживают два основных режима работы: обучение системы и проверка подлинности при доступе.

Основным достоинством описанных выше систем является простота построения. Широкие возможности их реализации на основе стандартных процедур цифровой обработки сигнала и невысокие требования к вычислительным ресурсам и объёму памяти ЭВМ сделали такие системы почти хрестоматийным примером при изучении теории автоматического определения человека по голосу.

Заключение. В рамках данной работы исследовались существующие методы решения задачи автоматической идентификации диктора по голосу.

Речь представляет собой сложный сигнал, образующийся в результате преобразований, происходящих на нескольких различных уровнях: семантическом, языковом, артикуляционном (уровне голосового аппарата человека) и акустическом (уровне физических свойств звука). Различия в этих преобразованиях влекут за собой различия в свойствах речевого сигнала. При решении задачи распознавания диктора по голосу все эти различия могут быть использованы для того, чтобы выделить индивидуальные характеристики голоса каждого человека.

Список цитируемых источников

1. Симончик К. К. Методы и алгоритмы автоматической текстонезависимой верификации дикторов и их программная реализация. СПб. : КопиСервис, 2010.
2. Машонский И. Д. Автоматическая идентификация диктора по голосу. М. : МГУ им. Ломоносова, 2014.
3. Там же.
4. Идентификация пользователей по голосу [Электронный ресурс]. URL: <http://www.speech-soft.ru/user-identification-by-voice?destination=node/14> (дата обращения: 04.02.2016).

УДК 004.02

А. Ю. Бузук, А. В. Максимович

Учреждение образования «Барановичский государственный университет», Барановичи

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ И ПРОЦЕДУРНОГО ПРОГРАММИРОВАНИЯ НА ПРИМЕРЕ РАЗРАБОТАННОГО ПРИЛОЖЕНИЯ

Введение. Сегодня существуют различные по сложности программы, что обуславливает программистов изучать и использовать тот или иной стиль программирования для создания своего приложения.

Выбор подхода для написания программ зависит от сложности технического задания, а иногда и от подготовки и выбора технологии программирования. Но всё же хотелось бы разобраться, что целесообразнее использовать объектно ориентированное программирование (далее — ООП) или процедурное программирование (далее — ПП).

Основная часть. Метод ООП основан на том, что любая программа состоит из объектов. Объект, в свою очередь, является экземпляром определённого класса. Классы позволяют проводить конструирование из полезных компонент, обладающих простыми инструментами, что даёт возможность абстрагироваться от деталей реализации. Данные и операции вместе образуют определённую сущность и они не «размазываются» по всей программе, как это нередко бывает в случае процедурного программирования. Локализация кода и данных улучшает наглядность и удобство сопровождения программного обеспечения. Инкапсуляция информации защищает наиболее критичные данные от несанкционированного доступа [1].

Стиль ООП даёт возможность создавать расширяемые системы. Это одно из самых значительных достоинств ООП, и именно оно отличает данный подход от традиционных методов программирования. Расширяемость

означает, что существующую систему можно заставить работать с новыми компонентами, причём без внесения в неё каких-либо изменений. Компоненты могут быть добавлены на этапе выполнения [2].

К недостаткам ООП можно отнести: необходимость понимать базовые концепции (класс, наследование, динамическое связывание); неэффективность в смысле распределения памяти (динамическое связывание и проверка типа на этапе выполнения требуют по ходу работы информацию о типе объекта, а такая информация храниться в дескрипторе типа, и он выделяется один на класс); излишняя универсальность (в библиотечном классе часто содержится больше методов, чем это реально необходимо) [3].

Метод ПП предоставляет возможность программисту определять каждый шаг в процессе решения задач. Особенность таких языков программирования состоит в том, что задачи разбиваются на шаги и решаются шаг за шагом. Используя процедурный язык, программист определяет языковые конструкции для выполнения последовательности алгоритмических языков [4].

Сравним использование двух методов на примере разработанного веб-приложения «Телефонный справочник», который предназначен для хранения данных об абонентах. В приложении было реализовано несколько веб-страниц, на каждой из которых отображается форма администратора для ввода данных и таблица для отчёта об абонентах сотовой и стационарной связи.

Представим основную форму приложения «Телефонный справочник» (рисунок 1).

Данное приложение выполняет следующие функции: 1) осуществляет определённую защиту и архитектуру (админ, гость); 2) имеет доступ к базе данных, в которой хранится справочная информация о телефонах и их владельцах; 3) удаляет данные и пополняет базу новыми данными; 4) осуществляет поиск, выполняя определённые запросы.

В ООП по сравнению с ПП программист представляет программу в виде набора классов. В приложении разработано 14 классов (рисунок 2).

Представим базовый класс `post` с используемыми методами, наследуемый от класса `My` (рисунок 3). Все остальные классы (`ADDServlet`, `GostServlet`, `LOGINServlet`, `MYFilter`, `MyServlet`, `PoiskServlet`, `ServletBystPoisk`, `ServletBystrPoisGlavn`, `ServletEdit`, `ServletNewWindow`, `Servletout`) являются классами-обработчиками.

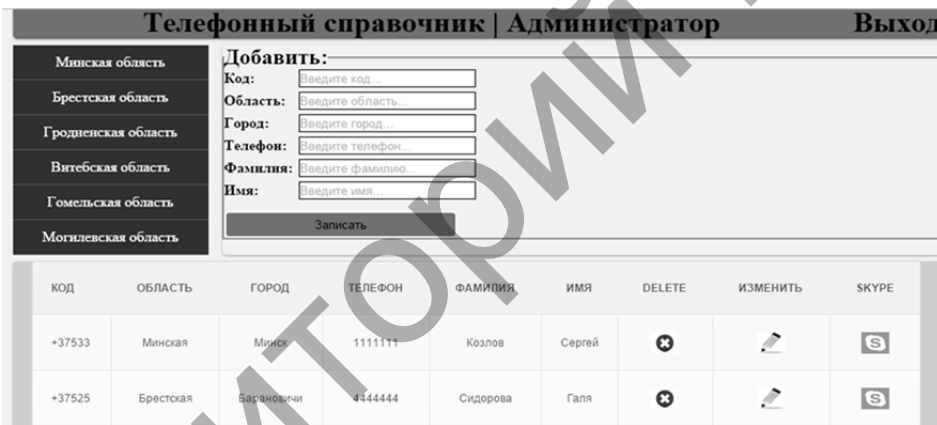


Рисунок 1 — Основная форма приложения



Рисунок 2 — Список классов

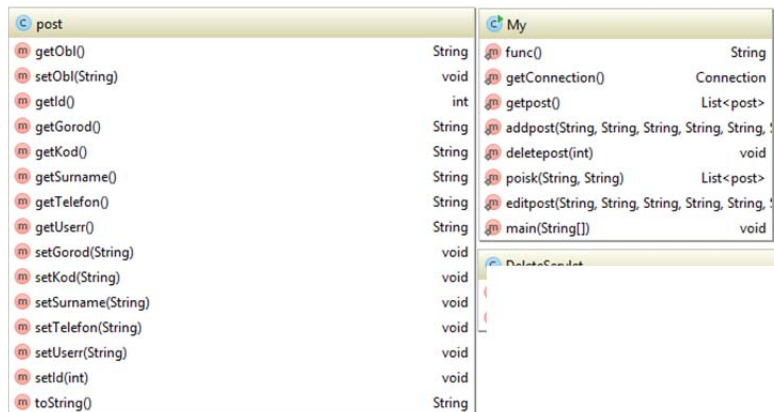


Рисунок 3 — Базовый класс `post`

Если бы при написании приложения использовалось ПП, то визуального отличия не наблюдалось. А вот во внутренней реализации программы всё масштабно поменяется: вместо классов — множество функций, которые и реализуют программу [5]. Например, для данного приложения пришлось бы разработать приблизительно 50 функций для обработки каждой формы ввода и вывода данных для абонентов сотовой и стационарной связи. А такое количество кода уменьшает производительность программного продукта и требует большого объёма памяти.

Заключение. Если имеется масштабная система, то без ООП не обойтись. Оно показывает отличные результаты в условиях изменения требований, обладая гибкостью в обслуживании и возможностью повторного использования кода. Кроме этого, проекты ООП легче поддаются автоматическому тестированию.

Но это совсем не значит, что не должно использоваться ПП. Если сначала оценивать, какой функционал требуется, т. е. создаётся прототип, то лучше начать с ПП. Пока не нужно продумывать всю архитектуру будущего приложения. Нужен минимальный функционал, чтобы примерно показать, как всё будет работать.

Список цитируемых источников

1. Влссидес Дж., Гамма Э., Хелм Р. Приёмы объектно ориентированного проектирования. СПб. : Питер, 2013. 368 с.
2. Там же.
3. Там же.
4. Прохоренко Н. А. Программирование на С++ в Visual Studio. М. : ДМК, 2000. 203 с.
5. Штрауб Б., Чакон С. Git для профессионального программиста. СПб. : Питер, 2013. 496 с.

УДК 004.65(476)

Е. Г. Вашкевич, М. Ю. Семашко

Учреждение образования «Барановичский государственный университет», Барановичи

ИСПОЛЬЗОВАНИЕ ОБЛАЧНЫХ ХРАНИЛИЩ ДАННЫХ В ПРЕПОДАВАТЕЛЬСКОЙ ДЕЯТЕЛЬНОСТИ

Введение. Облачное хранилище данных — это виртуальный носитель информации. На виртуальном диске возможно хранение информации и работы с ней не на нашем компьютере, а на сторонних серверах, которые разбросаны по всей сети Интернет и могут находиться на разных континентах. Облачное хранилище позволяет пользователям хранить и получать информацию (необходимые данные) в любом объёме и в нужный момент. Основным преимуществом таких хранилищ является то, что мы нагружаем не персональный компьютер или ноутбук, а используем возможности серверов компании, предоставляющей данный вид услуги. Мы же получаем лишь результат на наш компьютер через сеть Интернет. Наиболее известные из бесплатных облачных хранилищ данных — Google drive, Dropbox, One drive (ex. SkyDrive), Яндекс. Диск, Mega, а также почти все почтовые поисковые системы. В наше время данные облачные хранилища получили большое распространение, и уже сегодня их насчитывается несколько сотен, хотя ещё многие компании относятся к ним с некоторым недоверием.

Приятной особенностью «облачной» модели программных платформ является отсутствие необходимости в тщательном изучении системных требований, покупке дорогих комплектующих и постоянного совершенствования программного обеспечения. Все затраты на сохранность данных и функционирование программы берут на себя компании-разработчики. В обязанности компании-разработчика также входит обеспечение защиты и конфиденциальность информации, разработка программ для шифровки информации, наём и обучение персонала для администрирования серверов. Немаловажной проблемой конфиденциальности информации является проблема взаимоотношений с государственными структурами, проявляющими интерес к информации. Юристы компаний-разработчиков облачных хранилищ данных решают возникающие вопросы довольно успешно.

Получить доступ к информации можно с любого компьютера или ноутбука, для чего на сайте нужно создать учётную запись, всего лишь следуя требованиям предоставления услуги. После того, как это сделано, пользоваться информацией можно с любого персонального компьютера, ноутбука, планшета. Для доступа к информации достаточно интернет-подключения, логина и пароля.

Одними из самых распространённых хранилищ можно считать сервисы Google с большим количеством возможностей, которые используются для коллективной и индивидуальной работы. Сервисы Google удобны, понятны и доступны в использовании.

При использовании сервисов Google имеется возможность создания и редактирования документов, таблиц, презентаций; календарей и заметок, позволяющих вести планирование; интересных видеоматериалов; персональных веб-сайтов; карт и маршрутов.

Также возможно совместное редактирование и обсуждение в сети созданных документов и др.

Основная часть. Стремительное развитие новых информационных технологий оказало огромное влияние на существующую систему образования и повлекло её реформирование. Произошло появление новых областей