

**ШЕВЕЛЕВ А.С.**

*КНУ им. Ж. Баласагына, Бишкек*

**Shevelev A.S.**

*J. Balasagyn KNU, Bishkek*

[ac\\_asv@mail.ru](mailto:ac_asv@mail.ru)

## **МЕТОДИКА «ЕДИНОГО ПРОЕКТА» ПРИ ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ**

**«Биримдик долбоор» программированиясынын окутуу  
методикасы**

**«Unitary project» methods when getting education in  
programming**

***Аннотация:** в статье обобщены некоторые приемы преподавания предметов, связанных с программированием, которые применял автор на протяжении многих лет работы в Кыргызском национальном университете им. Ж. Баласагына. Методика основана на последовательном выполнении студентами единого проекта, связанного с применяемой системой программирования на протяжении одного семестра. Например, Delphi, 1С Предприятие, Visual C++. В статье также описываются простые приемы создания многовариантного проекта для обеспечения каждого студента своим вариантом.*

***Аннотация:** статьяга кээ бир ыкмаларды сабактарды үйрөтүү , программированиянын байланышы, аларды автор көптөгөн жылдар бою Ж. Баласагын атындагы Кыргыз улуттук университетинде колдонуп келген. Тургузулган методиканын катар аткарылышы студенттер окуу тобунун биримдик долбоору, берилген системанын байланышы программированиясы бир семестрдин узагында. Мисалы, Delphi, 1С Предприятие, Visual C++. Бул статьяда жөнөкөй мисалдардын түрдүү вариант проекти ар бир студенттин өз вариантын кабил алуу үчүн жазылган.*

***Annotatuon:** this article summarizes some of the teaching methods, related to the programming, that author used during long time of work in Kyrgyz National University, named after J. Balasagyn. The technique is based on the sequential execution by the students of training group of «unitary project», related for applied software system for one semester. For example, Delphi, 1C, VisualC++. This article also describes how to create simple methods of multivariate project for each student to their goal.*

***Ключевые слова:** программирование; информационные технологии; образование; обучение; методика; проект.*

***Негизги сөздөр:** программало; маалымат технологиялары; түзүү; үйрөтүү; методика; долбоору*

*Keywords: programming; information technology; education; training; technique; project.*

Автор, по образованию прикладной программист, хорошо помнит то время, когда учился в институте и получал первые навыки программирования на языке низкого уровня Assembler, а в дальнейшем - Алгол 60 и PL-1, для IBM 360.

В то время учили просто: давали список команд (операторов), объясняли их назначение и тренировали студентов применять эти команды для решения простейших задач, которые постепенно усложнялись. В то время компьютеры были в дефиците, поэтому студентам приходилось буквально «охотиться» за машинным временем. Это приводило к тому, что студент, буквально «вылизывал» свою программу, чтобы ошибок было меньше и чтобы за один подход к компьютеру можно было завершить задание. Дополнительный опыт приобретался некоторыми студентами при выполнении уже реальных задач по «хоздоговорным» работам. В результате такого обучения мы прилично знали языки программирования, которых было немного - два, три, и не более.

Можно ли в настоящее время обучать таким образом программированию? Ответ: да, можно. Это необходимо, но недостаточно.

Во-первых, количество языков чрезвычайно велико. В «облачном» сервисе для программирования Ideone, можно применять 77 языков [2, 3]. Данный сервис использовался в педагогическом эксперименте диссертационного исследования Чумаевой К.М. [4]

Во-вторых, и самое главное, изучить синтаксис языка – это далеко не основное. Язык программирования обычно «спрятан» внутри системы программирования, которая чрезвычайно сложна для человека, не связанного с компьютером.

Программисту приходится изучать выбранную систему программирования. И это достаточно трудоемкая работа, т.к. встроенный язык тесно связан с системой программирования. Кроме обычных конструкций, типа while, for, if, read, write и пр., добавляются правила работы с проектами, объектами, базами данных и т.д.

На протяжении многих лет автор преподавал предметы с разными названиями, но связанные с программированием. В процессе преподавания была выработана некоторая методика, которая, по личному мнению автора, достаточно эффективна. Подтверждением эффективности является количество выпускных бакалаврских работ, в

которых используются преподаваемые системы программирования. Это количество составляет более 50%. Следовательно, эти 50% студентов достаточно владеют системой программирования, пройденной на курсе.

Суть методики в следующем:

- Студентам предлагается выполнить один проект. Проект связан с понятными производственными задачами.
- В рамках проекта необходимо последовательно выполнять задания. Количество заданий соответствует количеству учебных часов.
- Каждое следующее задание обычно использует предыдущее.
- После выполнения всех заданий проект считается завершенным.
- Каждое задание надо сдать устно с демонстрацией на компьютере. За каждое задание начисляются баллы (по 20-бальной шкале)
- Для организации многовариантности проекта студенты обязаны добавлять свою фамилию к именам некоторых объектов. Преподаватель указывает в методических материалах перечень таких объектов. Это существенно препятствует механическому «списыванию» проекта друг у друга.
- В конце семестра суммарное количество баллов оказывает влияние на экзаменационную оценку в соответствии с системой оценивания, принятой в университете.

Как следует из сказанного, методика достаточно проста. Трудность заключается в разработке качественного методического материала по выполнению проекта.

Автор обычно практически не вмешивается в работу студентов на занятиях, позволяя общаться на занятиях и помогать друг другу. Студенты работают самостоятельно, используя лишь методические указания и встроенную справочную систему.

Методические указания перерабатываются каждый год с учетом замечаний студентов и новых версий систем программирования.

Далее для демонстрации методики автор прилагает некоторые реальные методические материалы, изложенные в сокращенном виде. Методика применялась для разных программных систем, в разные годы.

Прежде всего, содержание проекта связывается с деятельностью небольшой организации, выполняющей услуги. Данная идея

принадлежит разработчику курса программирования на платформе «1С Предприятие» Радченко М.К. [1].

Постановка задачи. Организация, **название** которой привязано к фамилии студента, выполняет **услуги**, например, ремонтирует электронные (или подобные) устройства. Работу для **клиентов** выполняют **мастера**. В своей работе используются различные **материалы**, например, детали, платы и пр.

В тексте выделены слова: название, услуги, клиенты, мастера, материалы. Это – будущие таблицы базы данных или объекты для некоторых систем (например, для 1С Предприятия).

Название организации – это название проекта. Т.к. оно привязано к фамилии студента, то все проекты отличаются названием. Даже этот простой прием очень затрудняет механическое копирование работ в современных системах программирования.

Приведенная выше постановка задачи автоматически влечет реализацию следующих минимальных функций:

1. Формирование скидок постоянным **клиентам**.
2. Формирование зарплаты **мастерам**, выполняющим **услуги**.
3. Формирование отчетов по расходу **материалов**.
4. Формирование остатков по **материалам**. Легко заметить совпадение выделенных слов с перечисленными в

постановке задачи.

Можно проиллюстрировать методику на двух системах программирования:

1. Borland Delphi 7.
2. 1С Предприятие 8.2. Borland Delphi 7. Для реализации данного проекта, требуется

некоторая начальная подготовка студентов. Поэтому проект предлагался на 2 курсе, во втором семестре, когда студенты уже получили первоначальные навыки по программированию на языке Pascal.

Предлагалось использовать базу данных Paradox, которая адаптирована для работы с Delphi. Базу данных автор предлагал называть по своей фамилии (рис. 1):

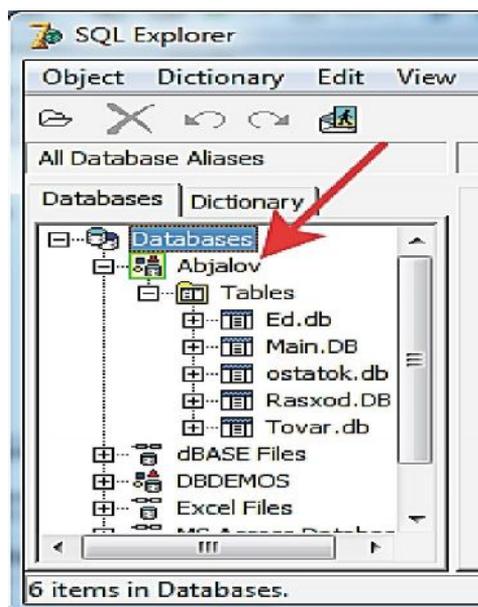


Рис.1. Пример использования фамилии для названия базы данных в Delphi.

Название некоторых таблиц, например, *материалы*, также предлагалось изменить с учетом фамилии. Например, NameMatAbjalov. Этот прием дает дополнительную вариантность проекту. Остальные названия можно было определять в соответствии с методическими указаниями.

Для реализации перечисленных выше функций в методических материалах для организации интерфейса и работы с таблицами базы данных предлагалось и описывалось, как использовать стандартные компоненты: MainMenu, Table, DBGrid, DataSource, DBNavigator.

Для связи разных таблиц при вводе, например материалов, использовался механизм подстановочных полей, через свойство поля Lookup.

Отчеты, например для формирования остатков материалов, предлагалось выполнить без применения стандартных компонентов вида QuickRep, а чисто программным способом, открывая нужные таблицы (номенклатуру, приход и расход), в цикле просматривая их и выбирая необходимую информацию. По мнению автора, это хорошая тренировка для улучшения программных навыков. Результат отчета должен быть примерно таким (рис.2):

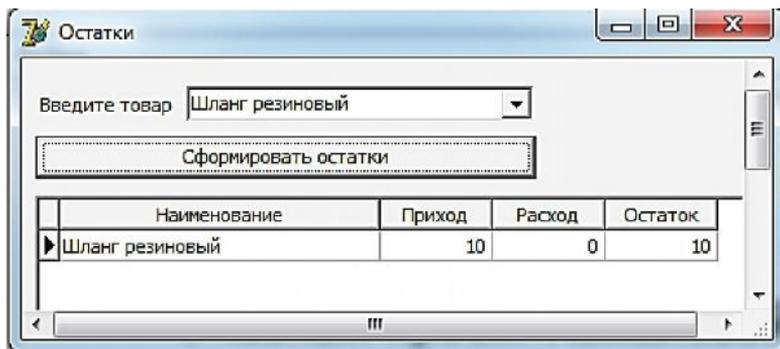


Рис.2. Пример формы отчета по остаткам материала в Borland Delphi 7

1С Предприятие. На этой системе программирования автор остановится более подробно, т.к., во-первых, платформа широко применяется во многих странах, включая Кыргызстан, а во-вторых, система существенно облегчает реализацию реальных производственных задач.

В данной системе программирования реализация проекта более приближена к постановке, чем Borland Delphi, Visual C++ или подобная.

В 1С нет базы данных как таковой. Система предлагает программисту работать с такими объектами, как справочники, документы, планами счетов, журналами и пр. Программист может создавать подобные объекты с разными названиями и разной структуры.

Каждый объект, например документ *Приходная накладная*, разрабатывается в конфигураторе - так называется среда разработки в 1С. Ее также называют платформой «1С Предприятие», которая различается версиями. На кафедре используется лицензионная учебная версия 1С Предприятие 8.2. (2016 год).

Пользователь работает не с конфигуратором, а с результатами конфигурирования. Внешний вид готовых решений 1С может отличаться по содержанию, но, по сути, очень узнаваемый. Один из вариантов приведен на рис. 3:

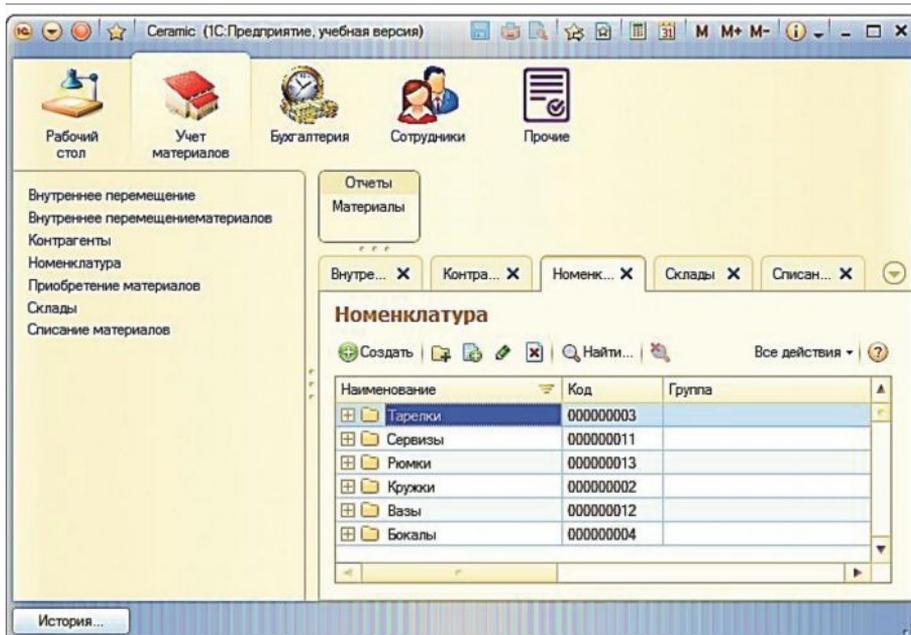


Рис. 3. Пример интерфейса для пользователя 1С версии 8.2 или 8.3

На данном рисунке показан внешний пользовательского интерфейса из курсовой студенческой работы на кафедре.

Легко заметить, что интерфейс включает: верхнюю панель инструментов, рабочий стол с главными объектами и список кнопок – подсистем (учет материалов, бухгалтерия, сотрудники, прочие). Текущее состояние рабочего стола запоминается.

Все это не требует программирования, в отличие от других систем программирования. Т.е. сам интерфейс формируется автоматически и может быть установлен стандартным. Но при желании его можно изменить.

Для получения подобного результата в Delphi, Visual C++ и даже в Microsoft Access надо прилагать некоторые усилия. Это обстоятельство отчасти объясняет выбор студентов, которые используют 1С при выполнении выпускных бакалаврских работ, а также магистерских диссертаций.

Таким образом, 1С имеет достаточно мощные визуальные средства проектирования прикладного решения. Но для его функционирования предлагается и встроенный объектно-ориентированный язык программирования.

Это полноценный развитой язык, который поддерживает все обычные языковые конструкции, а также приемы работы с прикладными объектами (документами, справочниками и пр.).

Особенностью языка является написание всех ключевых слов по-русски. При желании можно использовать и английский вариант написания: if, else, for, while и т.д. Допускается использовать два языка одновременно.

Еще одна особенность 1С представлена так называемыми синонимами всех имен прикладных объектов. Например, справочник имеет имя и синоним. Имя использует программист при написании кода, а синоним видит пользователь. И это имя можно писать на любом языке, на национальном, например. Аналогично эти свойства имеют все объекты, включая более детальные, например, реквизиты справочника.

Автор более подробно остановился на особенностях системы программирования 1С потому, что мало доступной информации на эту тему. В основном – это специальная литература.

И все же автор не рекомендовал бы платформу 1С для использования в обучении основам программирования. Слишком она отличается от стандартных систем программирования. Но для применения полученных ранее программных навыков она подходит вполне. Особенно на старших курсах.

Платформа 1С дает возможность решать реальные практические задачи, конечно в упрощенном виде, в процессе обучения. Тем более, что общие принципы современного программирования в 1С присутствуют – это объекты, свойства, методы, прерывания, события, обработчики событий и пр.

На протяжении нескольких лет автор использовал «проектный» подход в учебном процессе по предмету «Системы программирования» с использованием 1С на третьем курсе и магистратуре.

На первом занятии студенты должны создать новый проект (конфигурацию) с определенными свойствами, с использованием своей фамилии.

На втором занятии, в рамках проекта, разрабатывают **справочники**: сотрудники, клиенты и склады. Каждый справочник имеет свою особенность. Например, «Сотрудники» имеет табличную часть, «Клиенты» – стандартный справочник, а справочник «Склады» должен содержать в списке один особый склад - «Основной». В пользовательском режиме справочники заполняются данными.

На третьем занятии создается еще один справочник - «Номенклатура», как более сложный, т.к. он является иерархическим. Номенклатура состоит из материалов и услуг, которые детализируются. Заполнение справочника также выполняется в пользовательском режиме.

На четвертом и пятом занятиях разрабатываются **документы** «Приходная накладная» и «Оказание услуг».

Документы, в отличие от справочников, имеют свои особенности. Это наличие номера документа и даты создания. Документы также влияют на состояние учета в учебной организации. Например, приход увеличивает остатки по номенклатуре материалов. Поэтому для документов предусмотрена кнопка «Записать и провести», связанная с процедурой проводки. Эту процедуру программист должен написать на встроенном языке. Но в этих заданиях писать подобный код еще не требуется. Но коснуться языка программирования все же придется. Автоматический расчет суммы по количеству и цене студенты сделать должны.

На шестом занятии требуется сформировать выходную форму документов, которая может иметь различный вид. При этом появляется кнопка «Печать». Эта кнопка связана с процедурой печати, которая может формироваться без программирования через специальный конструктор.

На седьмом занятии предлагается поработать с проводками документов. Для этого уже требуется использовать элементы программирования. Данная теория приведена в методических материалах.

На восьмом - задание по формированию отчетов, которые основаны на результатах проводок документов. 1С имеет очень мощные механизмы создания отчетов, включая автоматические, без программирования, но включая работу с запросами. В 1С используется язык запросов, аналогичный SQL.

Дальнейшие занятия связаны с различными расчетами и отчетами: определение остатков на складе, расчет выручки сотрудников, которые выполняют услуги и т.д.

Все эти задания выполняются в рамках одного проекта. В результате в конце семестра студент получает работающую систему, в которой реализованы следующие функции:

- Ввод и корректировка сотрудников, клиентов, складов и номенклатуры.

- Ввод, корректировка и печать документов по приходу материалов и оказанию услуг.
- Выполнение расчетов и печать по остаткам материалов на складе.
- Выполнение расчетов и печать данных, необходимых для выдачи зарплаты.
- Выполнение расчетов и печать данных, необходимых для определения скидок клиентам.

Всего заданий 14, или 28 часов. Четыре часа используется для проведения модулей.

К сожалению не все студенты реализуют проект до конца. Отчасти потому, что задания взаимосвязаны. Невозможно выполнить задание №6, не выполнив №5. Конечно, студент может скопировать задание №5 у соседа. Но в заданиях обязательно присутствует требование использовать свою фамилию (рис. 4):

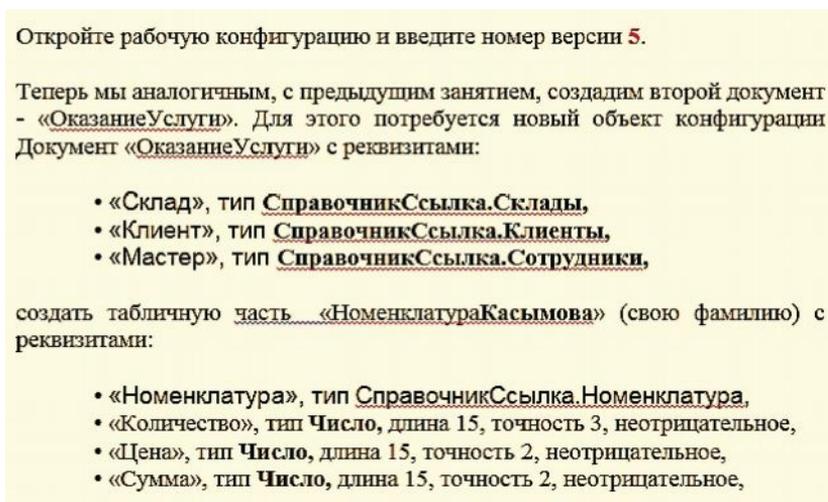


Рис.4. Фрагмент задания №5

Это препятствует автоматическому «списыванию», т.к. имена используются в коде программы (рис. 5):

```
«НаКлиенте
Процедура НоменклатураКасьмоваКоличествоПриИзменении (Элемент)
    // Вставить содержимое обработчика.
    СтрокаТабличнойЧасти = Элементы.НоменклатураКасьмова.ТекущиеДанные;
    РаботаСДокументами.РассчитатьСумму (СтрокаТабличнойЧасти);
КонечПроцедуры
```

Рис.5 Фрагмент модуля формы с фамилией в коде программы.

Эти места необходимо найти и правильно исправить, а это подталкивает студентов к работе со своим персональным проектом.

В целом, первые 6-8 заданий большинство студентов выполняют, а три-четыре студента из группы - выполняют полностью.

Методика «проекта» по ее сути используется и другими преподавателями. Но принцип сквозного проекта, который бы объединял все практические задания в единое целое, встречается не часто.

### *Литература*

1. М.Г. Радченко М.Г. «1С: Предприятие 8.0. Практическое пособие разработчика. Примеры и типовые приемы» [Текст] // М., 1С Паблишинг, - 2004. – С. 6-7
2. Облачный сервис для программирования Ideone / [Электронный ресурс]. – Режим доступа: <https://www.ideone.com> (дата обращения: 28.12.2016).
3. Чумаева К.М. «Облачные» технологии – что это [Текст] // Вестник КНУ имени Ж. Баласагына. – 2014. – Вып. 3. – С. 112–117.
4. Чумаева К.М., Нуржанова С.А. «Облачные» возможности для обучения основам программирования [Текст] // Известия кыргызской академии образования. – 2016. – Вып. 2. – С. 119–126.