

# ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

УДК 618.3

Б. Е. Стариченко,  
Е. Б. Стариченко,  
А. Д. Шеметова

## СОВЕРШЕНСТВОВАНИЕ ИНФОРМАЦИОННО- ТЕХНОЛОГИЧЕСКОЙ ПОДГОТОВКИ СТУДЕНТОВ НА ОСНОВЕ СИСТЕМНО-ОБЪЕКТНОГО ПОДХОДА

В статье обсуждается возможность построения курса информатики и информационных технологий на основе системно-объектного подхода. Освоение прикладных программных систем предлагается осуществлять посредством решения системных задач анализа-синтеза на объектной основе. Для овладения навыками алгоритмизации и программирования рекомендуется использовать встроенные в прикладные пакеты объектные среды программирования, что обеспечивает преемственность с пользовательским курсом, а также знакомство с современными технологиями программирования. Обосновывается применение предлагаемого подхода для обучения информатике студентов гуманитарных и технического вузов, а также учащихся колледжей и школьников.

*Ключевые слова:* системно-объектный подход, обучение информатике, объектная основа информационных технологий, встроенная среда программирования.

The possibility of computer science and information technology course construction on the basis of system and object approach is discussed in the article. Study of applied program systems is proposed to be carrying out as decision of system analysis-synthesis tasks on object basis. For algorithmization and programming training are offered to use the object environments built in applied packages that provides continuity with the user course, on the one hand, and acquaintance to modern technologies of programming – to another. Realization of proposed approach is possible for various levels of training to computer science – at school, college, humanitarian and technical higher education institutions.

*Keywords:* system and objective approach, training to computer science, object basis of information technologies, object programming environments built in applied packages.

**Введение.** Анализ программ информационно-технологической подготовки специалистов различных профилей показывает, что даже для студентов гуманитарных специальностей в рамках дисциплины «Математика

и информатика» предусматривается изучение основ алгоритмизации и программирования. Тем более обучение элементам программирования сохраняется в учебных программах естественнонаучных и инженерных специальностей. При этом немалые усилия направляются на изучение языка PASCAL, который не является объектным и при всех его исторических заслугах уже не применяется при разработке приложений. В то же время благодаря разнообразию и полноте прикладного программного обеспечения большинству современных специалистов сегодня не требуется создавать какие-либо программы для решения конкретных производственных задач на основе классических языков и систем программирования. Другими словами, изучение алгоритмизации и программирования в традиционном варианте, предусмотренном действующими образовательными стандартами и учебными программами, не отвечает потребностям профессиональной деятельности выпускников, с одной стороны, и уровню развития информационных технологий, с другой.

Практикуемые подходы к освоению прикладного программного обеспечения также имеют свои недостатки, поскольку обучение чаще всего сводится к знакомству с некоторым стандартным инструментарием и приемами его использования при создании и редактировании документа. При этом, как правило, не дается обобщение и не акцентируется внимание обучаемых на общих принципах построения документа и программного интерфейса, существующих в подавляющем большинстве современных прикладных программ. Не демонстрируются также возможности расширения стандартного инструментария и применения элементов автоматизации обработки данных. Следовательно, и «пользовательский» подход к освоению информационных технологий не раскрывает многих возможностей современных программных систем и не отвечает уровню развития технологий.

Таким образом, выявляется несоответствие между необходимостью эффективного применения будущими специалистами современных информационных технологий в профессиональной деятельности и недостаточной нацеленностью на это существующего содержания информационно-технологической подготовки.

Устранить указанное несоответствие, с точки зрения авторов настоящей статьи, можно в рамках универсальной для многих специальностей (исключая, безусловно, специалистов в области IT) схемы информационно-технологической подготовки, основанной на двух положениях:

- последовательном применении системно-объектного подхода к построению документа и интерфейса программных систем;
- использовании встроенной объектной среды программирования.

Рассмотрим эти положения подробнее.

**Суть объектного подхода к описанию систем.** Одним из основных методологических подходов, используемых в современном научном и технологическом знании, следует считать системный подход. В разное время многие исследователи всесторонне изучали вопросы системного под-

хода и системного анализа: А. Н. Аверьянов, В. Г. Афанасьев, И. В. Блауберг, В. П. Кузьмин, Т. Кун, В. Н. Садовский, А. И. Уемов, В. А. Штофф, Э. Г. Юдин и др. Ими показана универсальность этого подхода при описании процессов в природных и общественных системах, а также продемонстрирована возможность применения его в качестве методологии научного познания мира.

А. Н. Аверьянов выделяет следующие аспекты, лежащие в основе системного познания мира:

- рассмотрение объекта деятельности (теоретической и практической) как системы, т. е. как ограниченного множества взаимодействующих элементов;
- определение состава, структуры и организации элементов и частей системы, обнаружение главных связей между ними;
- выявление внешних связей системы, выделение среди них главных;
- определение функции системы и ее роли в ряду других систем;
- анализ диалектики структуры и функции системы;
- обнаружение на этой основе закономерностей и тенденций развития системы [1, с. 9].

Познание мира вообще и научное в частности не может осуществляться хаотически, беспорядочно, оно имеет определенную структуру и подчиняется определенным закономерностям. Эти закономерности определяются закономерностями развития и функционирования объективного мира.

В работе Ю. А. Урманцева указывается, что использование системного подхода приводит к принципиально новым технологиям получения знания и системной организации практической деятельности, а именно:

- представлению любого объекта в виде объекта-системы, что неизбежно сопровождается эксплицированием законов о его первичных элементах, отношениях единства, законах композиции. Нередко такие знания сами являются открытиями (например, протонов, электронов, нейтронов в атомах);
- построению результирующей системы (Р-системы). Стихийные и сознательные построения Р-систем, как показывает история развития науки, подытоживают результаты предшествующих этапов развития данной отрасли знания, существенно обогащая ее [5, с. 62];

• современные требования информационного общества к человеку, его информационной культуре обусловили необходимость обучения новым методам познания, описания и изменения мира, и в частности системному подходу и системному анализу, которые в этом отношении универсальны и высоко эффективны. Такое обучение предполагает освоение базовых теоретических положений – понятий, подходов, умений выделять и описывать систему, ее компоненты и связи между ними. Безусловно, в каждой конкретной сфере научного или практического знания изучаются и создаются свои системы и, соответственно, в рамках общего подхода предлагаются его конкретные реализации. Вместе с тем, нам представля-

ется весьма оправданным и перспективным применение системного подхода при обучении современным информационным технологиям.

Рассмотрим некоторые положения названного подхода. Согласно общей теории систем, любую сложную реальную сущность можно рассматривать как систему. Не вдаваясь в терминологический анализ, мы примем следующее определение системы – это *совокупность взаимодействующих компонентов, каждый из которых в отдельности не обладает свойствами системы в целом, но является ее неотъемлемой частью* [3, с. 296].

Одним из способов представления системы является *объектный* способ.

*Объект* – простейшая составляющая сложного объединения (системы) – обладает следующими качествами:

- в рамках данной задачи не имеет внутреннего устройства и рассматривается как единое целое;
- имеет набор свойств (атрибутов), которые изменяются в результате внешних воздействий;
- идентифицирован, т. е. имеет имя (название).

Прокомментируем предложенное определение.

«*В рамках данной задачи...*» означает, что одна и та же сущность в одних задачах может рассматриваться в качестве простой (т. е. как объект), а в других – нет.

Объекты, безусловно, могут обладать свойствами. Понятие свойства определим следующим образом: *свойством (атрибутом) называется качество объекта, для которого установлена мера; сама мера называется значением атрибута.*

Набор свойств объекта обозначим как *поле свойств*. В поле свойств выделяются две составляющие: свойства индивидуальные и общие. К индивидуальным свойствам относятся те, которые отличают данный объект от множества ему подобных. *Класс* – это множество объектов, обладающих одним или несколькими одинаковыми атрибутами, которые называются *полем свойств класса*.

Свойства объекта могут изменяться с течением времени. Изменение свойства – это процесс. Любой процесс имеет причину («движущую силу»). По отношению к объекту причины протекающих процессов могут быть только внешними, поскольку, согласно определению, внутреннее устройство и, соответственно, какие-либо внутренние воздействия и причины у объекта отсутствуют. Внешние воздействия (причины) могут носить постоянный (непрерывный во времени) характер (например, притяжение к Земле) или быть дискретными – в этом случае их называют *событиями* (например, толчок тела или поступление порции информации). Реакцией объекта на внешнее воздействие является изменение его свойств.

Описание любого объекта начинается с присвоения ему идентификатора, т. е. имени, – без него невозможно указать, какая сущность рассматривается. *Имя* (название) *объекта* является его индивидуальным признаком, который, однако, нельзя считать свойством, поскольку оно не имеет

меры. *Имя* (название) *класса* является общим признаком для группы объектов. Имя объекта или класса не может изменяться с течением времени; изменение имени (переименование) следует рассматривать как прекращение существования одного объекта (класса) и возникновение другого.

На практике необходимость выделения и описания систем связана с постановкой и решением следующих задач:

- *изучение прототипа системы*, т. е. выяснение строения (компонентов) ее природного или искусственного прототипа, особенностей связей между компонентами, влияния внешних и внутренних факторов на характер протекающих процессов;

- *описание системы*, т. е. представление языковыми или графическими средствами;

- *построение системы* – создание новой системы из компонентов;

- *использование системы* – решение с ее помощью каких-то практических задач.

При решении перечисленных системных задач используются два метода – анализ и синтез.

*Анализ* – метод исследования, основанный на выделении отдельных компонентов системы и рассмотрении их свойств и связей. Анализ – это декомпозиция (расчленение) сложного объединения на составные части и изучение их свойств и связей между ними по отдельности.

*Синтез* – 1) метод исследования (изучения) системы в целом (т. е. компонентов в их взаимосвязи), сведение в единое целое данных, полученных в результате анализа; 2) создание системы путем соединения отдельных компонентов на основании законов, определяющих их взаимосвязь. Синтез – это объединение составляющих для получения нового качества (системного свойства). Такое объединение возможно только после изучения свойств компонентов и закономерностей их взаимодействий, а также влияния различных факторов на системные свойства. Синтез – целенаправленная деятельность человека, следовательно, его результатом будет *искусственная* система (в отличие от природной – *естественной*).

### **Объектная основа современных информационных технологий.**

Наиболее последовательно объектный подход реализован в современных программных системах. Ключевыми оказываются представления об экранных объектах, их свойствах и инструментах, с помощью которых свойства можно изменять. В этом случае освоение любой объектной прикладной программы сводится к выявлению объектов, которые поддерживаются программой, их свойств, существующих инструментов для изменения свойств и порядка доступа к ним, а также характера взаимодействия данного экранного объекта с остальными. Другими словами, программа рассматривается как инструментальная система, которая позволяет создать другую систему – документ, представляющий собой совокупность взаимосвязанных объектов. Следовательно, при освоении программных систем решается системная задача анализа. Подготовка документа в некоторой инструментальной среде представляет собой синтез документа-системы из имеющихся объектов. Та-

ким образом, освоение любых объектных программных систем может осуществляться по единой (универсальной) схеме:

- выявить объекты, поддерживаемые средой, и их свойства;
- подобрать инструменты, управляющие свойствами объектов, и порядок доступа к ним;
- обнаружить характер взаимодействия объекта с другими объектами, а также инструменты, позволяющие изменять этот характер.

Следует отметить, что многие современные программы имеют унифицированный интерфейс – некоторый набор инструментов присутствует обязательно (и, как правило, они имеют одинаковые обозначения), инструменты располагаются группами на панелях или в меню, назначение инструментов имеет контекстные подсказки, доступ к одному и тому же инструменту может осуществляться по-разному, что позволяет пользователю выбрать наиболее удобный для него вариант.

Таким образом, важными особенностями реализации объектного подхода в прикладных программах оказываются следующие:

- из всевозможных объектов выделяется некоторое множество инвариантных объектов – тех, что присутствуют в различных программах; это позволяет унифицировать их интерфейс (например, кнопки, окна, панели, меню и др.);
- освоение каких-либо объектных программ сводится к изучению интерфейса объектов; общий же принцип – документ конструируется из объектов – одинаков для всех программ;
- для начала работы с программой не требуется полного изучения всех ее объектов и инструментов – достаточно освоить некоторые минимально необходимые возможности, а остальные – по мере востребованности.

К числу преимуществ объектного подхода следует отнести: единую технологию формирования документа как совокупности экранных объектов; событийный характер изменения свойств объектов, реализуемый посредством набора инструментов; возможность осуществления экспорта-импорта объектов между различными приложениями (OLE и DDE-технологии); дружественный интерфейс пользователя, построенный по одинаковым принципам для различных программ (что создает предпосылку для их самостоятельного освоения); доступ к использованию встроенной объектно-ориентированной среды программирования для расширения стандартных возможностей базового прикладного пакета.

Благодаря освоению систем объектного программирования можно не только пользоваться готовыми конструкциями, но и создавать собственные объекты или классы, а воздействие на объект описывает, применяя названия (имена) методов, а не визуальные инструменты. Разработанная программа изначально также имеет объектную основу и до компиляции представляет собой некоторый объектный экраный документ. Правда, после компиляции он становится исполняемой программой, которая, в общем случае, существует самостоятельно и, в отличие от документа, не привязана к породившей ее инструментальной среде.

Таким образом, оказывается возможным основанное на единых и универсальных идейных позициях целостное построение курса информатики любого уровня.

**Системно-объектный подход в пользовательском курсе информатики.** Рассмотрим более детально методические аспекты реализации базового (пользовательского) курса информатики, построенного на системно-объектной основе. В освоении курса участвовали различные группы обучаемых – школьники, студенты колледжа, студенты педагогического и технического вузов. Безусловно, глубина и степень детализации положений были различны, однако общая последовательность изложения оставалась инвариантна для всех.

Обучение информатике в рамках нашего курса начинается с теоретического модуля, в рамках которого формируется система базовых понятий системно-объектного подхода и обсуждаются универсальные схемы решения информационных задач [4]. Нам представляется логичным и методически оправданным следующий порядок введения базовых понятий:

- на доступных примерах формируются понятия *сложное* (состоит из чего-то) и *простое* (не имеет строения), а также представление об их относительности;

- вводится понятие *модели* как упрощения прототипа, отражающего его существенные для данной задачи свойства; обсуждается множественность моделей одного прототипа, основные классы моделей, а также виды моделей информационных;

- понятие *объект* устанавливается как модельное представление;

- необходимость описания свойств объектов приводит к понятиям *величина* и *тип величины*, а также к обсуждению представления простых и сложных типов данных в компьютере;

- обсуждаются особенности информационных программных объектов: объект объединяет *свойства* и *методы*; свойства выражаются значениями параметров; параметры могут иметь один из трех типов – числовой, символьный или логический; термин «методы» можно последовательно ввести позднее в разделе «Элементы программирования», а при изучении программного обеспечения достаточно использовать терминологию *событие* и *реакция объекта на событие*;

- вводятся понятия *класс объектов*, *признаки класса*, *классификация*: любой объект является представителем некоторого класса, в связи с чем какие-то свойства объекта оказываются общими для всего класса, а остальные – свойствами индивидуальными; классификация есть отнесение объекта к классу по некоторому признаку (или группе признаков); рассматриваются способы представления классификации (список, таблица, схема) и примеры классификаций; здесь же поясняется и иллюстрируется на примерах понятие «наследование свойств»;

- осуществляется ознакомление с понятием *процесса* как изменения свойств объекта в результате внешних воздействий; делением воздей-

ствий на непрерывные и дискретные; возможностью дискретизации воздействий; понятием *реакция объекта* на воздействие;

- отдельно обсуждается понятие *черного ящика* как конструкции с неизвестным устройством (по сути – объектом), но с известными реакциями на воздействия;

- вводится понятие *системы* как совокупности взаимодействующих компонентов, каждый из которых в отдельности не обладает свойствами системы в целом, но является ее неотъемлемой частью; подчеркивается, что выделение системы происходит в результате упрощения, т. е. система – модельное представление;

- обозначается, что компоненты системы могут быть двух типов: сложные – подсистемы, простые – объекты; таким образом, выделение объектов и связей между ними – конечный этап декомпозиции системы;

- обсуждаются основные подходы к решению системных задач: анализ и синтез, а также группы задач: изучение и описание прототипов систем, создание и использование систем; таким образом, системный подход оказывается представлен в качестве основной и универсальной методологии решения задач анализа систем, а объектный – задач синтеза (создания) систем;

- вводится понятие *формальной системы* и обсуждается роль формализации в процессе решения задачи на компьютере, а также общая последовательность решения;

- *алгоритм* рассматривается как частный случай формальной системы, обеспечивающей однозначное преобразование информации (данных); акцентируются именно алгоритмы обработки информации и управление процессом обработки;

- в связи с задачей описания алгоритма (формальной системы) вводится понятие *языка*, в частности формального.

После формирования перечисленного комплекса понятий большинство остальных вопросов, традиционно включаемых в курс информатики, могут изучаться как примеры применения общего системно-объектного подхода. Задачи анализа (изучение, описание) должны начинаться с выделения системы, ее декомпозиции до уровня объектов, выяснения их свойств и реакций, а также связей с другими объектами. Задачи синтеза (создание) начинаются со знакомства с имеющимся набором объектов, их свойствами и реакциями, а также правилами объединения и создания новых объектов.

Непосредственному освоению программных систем предшествовало теоретическое введение со следующей логикой освещения вопросов:

- при использовании компьютера для создания некоторого продукта (в широком значении термина – текста, рисунка, мультимедиа-продукта, звукового файла и пр.) мы имеем дело с двумя системами: инструментальной средой (приложением) и документом; эти системы определенным образом соотносятся между собой: приложение является исполняемой программой, документ – продуктом работы с ней, т. е. документ носит



подчиненный по отношению к программе характер; приложение может породить множество документов одного типа;

- системы – приложение и документ – различаются набором объектов и действий с ними; в объекты приложения входят окна, меню с группами команд, панели инструментов, панели настроек среды – в совокупности они образуют интерфейс приложения; реакцией на вызов объекта-инструмента является выполнение некоторой команды по преобразованию объекта (или группы объектов) в документе; важно подчеркнуть, что вызову инструмента обязательно предшествует указание (выделение) объекта (или группы объектов) документа, над которым осуществляется преобразование; преобразование состоит в изменении свойств объекта;

- вызов инструмента может быть произведен одним из трех способов: посредством представляющего его интерфейсного объекта (кнопки, пиктограммы), выбором из списка (меню), нажатием клавиши (или их сочетания) на клавиатуре; вызов многих команд для удобства пользователя дублируется несколькими – двумя и более – способами; вызов экранного инструмента производится при наведении на него указателя мыши с последующим щелчком левой клавиши;

- над объектом документа возможны следующие манипуляции мышью:
  - *активизация* – однократный щелчок по объекту левой клавишей мыши или «проводка» при нажатой левой клавише;

- *перетаскивание* – после выделения «захват» объекта левой клавишей мыши, перемещение на новое место экрана, освобождение клавиши;

- *изменение размеров* (для внедренных объектов) – после активизации объекта захват мышью одной из «горячих точек» и перемещение в нужном направлении;

- *вызов контекстного меню*, обеспечивающего доступ к инструментам данного объекта, – щелчок по объекту правой клавишей мыши;

- доступ к некоторым инструментам осуществляется посредством так называемых «окон диалога», в частности, это операции с внешним носителем (сохранение документа на носителе, считывание с него, вставка внешнего объекта из файла на носителе), вывод на печатающее устройство и передача по сети; в окне диалога следует сначала указать (выбрать) путь к папке, где документ будет храниться или откуда он считывается; при чтении – выбрать имя файла, в котором сохраняется документ, при записи – ввести имя и тип (расширение) файла; окна диалога одинаковы во всех приложениях данной операционной системы;

- операции копирования, вырезания и перемещения могут происходить с использованием *буфера обмена* – специально выделенного места на жестком диске или в ОЗУ компьютера для временного хранения данных; запись и хранение в буфере осуществляется в виде файла данных, имя которого и расположение являются одинаковыми для всех приложений – это позволяет осуществлять перенос данных из одного места документа в другое или между документами (даже разных приложений).

Рассмотрение перечисленных вопросов призвано сформировать у обучаемых обобщенное представление о построении приложений и документов, элементах интерфейса, приемах работы. Помимо этого обучаемые запоминают некую универсальную последовательность действий по освоению какой-либо объектной инструментальной среды:

- выяснить, из каких объектов строится документ, и каковы их свойства;
- определить, какими инструментами свойства могут быть изменены и каким образом осуществляется доступ к инструментам;
- выявить особенности взаимодействия объектов одного документа.

Дальнейшее изложение строится по методу «от общего к частному» – описанный обобщенный подход применяется при освоении конкретных приложений.

**Применение встроенных сред при обучении программированию.** Тема «Алгоритмизация и основы программирования» остается одной из центральных практически в любом курсе информатики (включая гуманитарные специальности). Однако достаточно часто у обучаемых возникают заметные трудности при ее освоении, поскольку за весьма ограниченное время им требуется изучить язык программирования с формализованным синтаксисом, а также приемы работы в среде программирования. В связи с этим актуальными оказываются поиски методических подходов, которые, с одной стороны, позволяли бы выполнить требования образовательных стандартов и познакомить учащихся с основными идеями и элементами программирования, а с другой стороны, обеспечивали бы логическую связь и преемственность с освоенным ранее пользовательским программным обеспечением. Возможным вариантом решения проблемы, с нашей точки зрения, является объединение пользовательской и программистской ветвей прикладной информатики на основе встроенной в прикладные пакеты объектно-ориентированной среды программирования. В экспериментальной части нашего исследования использовалась объектная среда Visual Basic for Applications (VBA) для пакетов MS Word и MS Excel.

Под *встроенной средой программирования* (ВСП) понимается *объектная среда, входящая в состав пакета прикладных программ и предназначенная для расширения функций базового пакета и автоматизации обработки документа.*

Встроенная среда программирования – это полноценная инструментальная система, обладающая качествами и возможностями систем объектного программирования. Основные приемы работы в ней такие же, как, например, в средах Delphi или Visual Basic. Вместе с тем имеется ряд специфических особенностей, позволяющих выделять ВСП в самостоятельный класс систем программирования:

- продукт, созданный с помощью ВСП, не относится к исполняемым (независимым) файлам – он работает лишь в рамках базовой среды; по этой причине ВСП является транслятором, в нем не предусмотрена компиляция конечной программы;

- объектами и методами ВСП служат объекты и инструменты базовой среды, что намного упрощает программирование, поскольку даже сложные методы (например, установка защиты на лист или скрытие документа) уже реализованы; при этом предусмотрена возможность описания новых классов и объектов;

- среда содержит инструмент *макрорекодер*, который позволяет фиксировать и запоминать некоторую последовательность команд по обработке документа в виде программного кода (*макроса*); пользователь может не быть знаком с программированием – формально он записывает некоторую цепочку операций (подобно записи на магнитофон) в поименованный макрос; в дальнейшем по имени макроса нужная последовательность действий может быть вызвана и воспроизведена.

При наличии соответствующих умений и знании языка VBA пользователь может просмотреть программу, записанную в виде макроса, и при необходимости ее дополнить или отредактировать.

Можно сказать, что применение ВСП позволяет устранить отмеченную выше нестыковку между пользовательской и алгоритмической составляющими курса информатики и информационных технологий – изучение элементов программирования всеми категориями пользователей оказывается мотивированным тем, что человек, изначально познакомившийся с некоторой объектной прикладной средой, получает возможность освоить новые приемы работы в рамках той же среды. Тем самым обеспечивается содержательная преемственность между пользовательским и алгоритмическим разделами.

Нами апробирована следующая методика изучения алгоритмизации и программирования с помощью ВСП [6].

Получив предварительную пользовательскую подготовку по описанной выше методике, обучаемые, используя знакомые им объекты текстового документа (символ, слово, абзац и т. п.), записывают последовательности команд средствами макрорекодера. Таким образом, могут быть реализованы простейшие задачи на изменение свойств объектов, например: изменить кегль шрифта каждого третьего слова в первом абзаце, вставить символ после каждого пятого слова в третьем абзаце и т. п. На этом этапе вводится понятие алгоритма, исполнителя, системы команд исполнителя и формального языка. Конкретным исполнителем оказывается сам текстовый редактор, объектами обработки – объекты документа. Подход к понятию «язык программирования» осуществляется с помощью записи на естественном языке последовательности действий, реализуемых затем в макросах. От естественного языка можно перейти к сокращенной записи команд – фактически учащиеся предлагают свою формализованную нотацию записи команд.

В дальнейшем устанавливается понятие среды программирования и языка программирования на примере встроенного редактора макросов и языка Visual Basic for Applications. На этом этапе осуществляется следующее:

- вводится понятие «событие» и рассматриваются реакции известных учащимся объектов на события;

- вводится понятие «метод» и описываются методы, применимые к различным объектам;
- внедряются новые, ранее не встречавшиеся, объекты (форма, окно, кнопка и др.);
- ставятся и рассматриваются задачи, требующие минимального редактирования макросов, полученных с помощью макрорекодера;
- приводится описание объектов и стандартных алгоритмических структур (цикл, ветвление и пр.) средствами VBA;
- решаются усложненные варианты задач первой части, а также задачи, решение которых невозможно без непосредственного программирования средствами языка (например, применить ко всем абзацам произвольного текста то или иное оформление, в зависимости от первого слова или имеющегося оформления текста).

Мы полагаем (и убедились в этом на практике), что рассматриваемый подход имеет следующие достоинства:

- позволяет вернуться к изначальному смыслу понятия «алгоритм» как *последовательности действий по обработке дискретной* (и, следовательно, знаковой) *информации*, который был утрачен в школьных курсах информатики в связи с применением исполнителей типа «паркетчик», «чертежник», «черепашка» и др. – они не связаны напрямую с обработкой информации, и корректнее было бы вести речь об *управлении* действиями объектов;

- знакомит учащихся непосредственно с основами современной объектно-ориентированной парадигмы программирования, позволяя миновать традиционное процедурное программирование и не тратить силы на освоение сложных сред типа Delphi или C++;

- сохраняет преемственность и естественную связь с изученными ранее офисными программами, что делает более наглядным и простым освоение базовых понятий алгоритмизации и программирования; помимо этого, безусловно, повышает уровень пользовательского владения прикладным программным обеспечением;

- позволяет осуществить уровневую дифференциацию учащихся по их возможностям и интересу к решению задач программирования. Минимальным и обязательным для всех является освоение описанного выше ядра. Для более «продвинутых» учащихся может быть предложено самостоятельное написание сложных макросов, рассмотрение стандартных и создание собственных алгоритмов решения задач, создание собственных объектов на базе имеющихся. В этом случае вводятся такие понятия объектно-ориентированного программирования, как полиморфизм, инкапсуляция, классы объектов. Желаящим можно предложить самостоятельно освоить аналогичный аппарат табличного процессора Microsoft Excel и использовать его при решении математических задач.

Таким образом, существует способ сделать освоение основ алгоритмизации достаточно «безболезненным» и интересным для всех учащихся. При этом глубина освоения алгоритмизации и программирования может

легко варьироваться в зависимости от направления подготовки. Например, студенты гуманитарного профиля могут ограничиться освоением программирования на основе макросов – в качестве продолжения и развития общей пользовательской подготовки. Студенты технических специальностей, учебным планом которых предусмотрено изучение программирования как самостоятельной дисциплины, освоив в базовом (пользовательском) курсе работу с макросами, получают возможность в рамках того же инструментария изучить объектный язык программирования.

**Организация и результаты эксперимента.** Нами были проведены исследования с целью изучения возможности и эффективности построения курсов информатики и программирования на основе описанного выше системно-объектного подхода. Исследование проводилось в ОТИ МИФИ (Озерском технологическом институте (филиале) Московского инженерно-физического университета) со студентами специальности 653700 Информационно-измерительная техника и технологии в течение двух учебных лет (2007/08 и 2008/09). Общий охват обучаемых, участвовавших в опытно-поисковой работе, составил 120 человек.

На заключительных этапах обучения студентам предлагалось выполнить индивидуальный профессионально-ориентированный проект, который впоследствии подвергался группой экспертов поэлементному анализу.

Было обосновано выделение показателей и критериев результативности, а также разработана процедура измерения этих показателей и интерпретации их значений. В качестве показателей результативности были выбраны:

- средние по учебной группе доли усвоения студентами отдельных элементов знаний и умений в соответствии с целями обучения встроенным средам программирования;
- индивидуальные доли усвоения материала в целом;
- различие средних групповых показателей (или отсутствие такового) для двух лет наблюдения.

Было установлено, что средние по группе согласованные экспертные оценки практически всех элементов (интерфейс проекта, оптимальность использования алгоритмических конструкций, знание основных классов и объектов MS Office, применение диалоговых окон посредством функций InputBox, Msg, использование новых команд и собственных надстроек, устойчивость к сбоям (ошибкам) и др. (всего 15 элементов)) превысили пороговое значение 70% в модели полного усвоения знаний В. П. Беспалько [2, с. 58], что, безусловно, свидетельствует о высокой результативности предложенной и апробированной методики обучения.

Стабильность средних групповых показателей для двух лет проверялась с помощью  $t$ -критерия Стьюдента. Анализ приведенных данных позволяет заключить, что отсутствует статистически достоверное различие средних групповых показателей ( $t_{\text{эксп}} = 1,76$ ,  $t_{\text{кр}} = 2,14$ ;  $t_{\text{эксп}} < t_{\text{кр}}$ ). Это говорит о стабильности результатов при использовании разработанной методики обучения.

Таким образом, в процессе исследования полностью подтвердилась исходная гипотеза о возможности освоения методов программирования на основе системно-объектного подхода и ориентации на использование встроенных сред программирования.

**Заключение.** Авторы статьи стремились показать, что применение системно-объектного подхода позволяет построить преподавание информатики и информационных технологий для всех категорий обучаемых на основе единых идейных позиций. Весьма важным представляется то обстоятельство, что оказываются логически связанными разделы, в которых осуществляется пользовательская подготовка к работе с основными программными системами, и разделы, посвященные изучению алгоритмизации и элементам программирования. При этом степень углубления в вопросы программирования гибко устанавливается в соответствии с программой дисциплины и потребностями будущей профессии. Экспериментальные данные, полученные в работе с различными категориями обучаемых, подтверждают возможность и справедливость представленного подхода.

#### Литература

1. Аверьянов А. Н. Системное познание мира. М.: Политиздат, 1985. 263 с.
2. Беспалько В. П. Слагаемые педагогической технологии. М.: Педагогика, 1989. 192 с.
3. Стариченко Б. Е. Теоретические основы информатики. М.: Горячая линия-Телеком, 2003. 312 с.
4. Стариченко Е. Б. Системно-объектный подход к проектированию и реализации курса информатики в колледже: дис. ... канд. пед. наук. Екатеринбург, 2003. 144 с.
5. Урманцев Ю. А. Системная философия (пять этюдов) // Вестн. Моск. ун-та. Сер. 7. 1999. № 5. 110 с.
6. Шеметова А. Д. О преимуществах изучения встроенных языков программирования в курсе информатики студентами специальности «Информационно-измерительная техника и технологии» // Актуальные вопросы методики преподавания математики и информатики: сб. науч. тр. Второй всерос. науч.-практ. конф. Биробиджан, 16 апр. 2007 г. Биробиджан: Изд-во ДВГСГА, 2007. С. 156–158.