

CASE-ТЕХНОЛОГИИ КАК СРЕДСТВО РЕАЛИЗАЦИИ КОМПЕТЕНТНОСТНОГО ПОДХОДА В ПОДГОТОВКЕ ПРОФЕССИОНАЛЬНЫХ ИНЖЕНЕРОВ-ПРОГРАММИСТОВ

Манжикова С.Ц., Десятков Г.А.

Кыргызско-Российский Славянский Университет, Кыргызская Республика,
msvetlana_88@mail.ru, gendes@krsu.edu.kg

Формирование профессионального мышления будущих инженеров программистов в условиях современного ВУЗа требует инновационного подхода, основной составляющей которого является компетентностный подход [1,2]. Идея и термин «компетентности» появились в 70-е годы XX века и широко использовались в США и в ряде стран Европы в связи с возникшей тогда проблемой индивидуализации обучения.

В Болонской декларации выделено несколько групп компетенций:

I. Общие компетенции. К ним относятся:

1. Инструментальные компетенции, которые включают когнитивные способности, способность понимать и использовать идеи и соображения; методологические способности, способность понимать и управлять окружающей средой, организовывать время, выстраивать стратегии обучения, принятия решений и разрешения проблем; технологические умения, умения, связанные с использованием техники, компьютерные навыки и способности информационного управления; лингвистические умения, коммуникативные компетенции.

2. Межличностные компетенции, то есть индивидуальные способности, связанные с умением выражать чувства и отношения, критическим осмыслением и способностью к самокритике, а также социальные навыки, связанные с процессами социального взаимодействия и сотрудничества, умением работать в группах, принимать социальные и этические обязательства.

3. Системные компетенции, то есть сочетание понимания, отношения и знания, позволяющее воспринимать, каким образом части целого соотносятся друг с другом и оценивать место каждого из компонентов в системе, способность планировать изменения с целью совершенствования системы и конструировать новые системы.

II. Специальные (профессиональные) компетенции.

Они определяют профессиональные требования к уровню знаний и степени подготовки бакалавров и магистров в соответствии с Государственными образовательными стандартами.

В связи с тем, что в современной педагогической науке и практике понятиями-категориями «компетенция» и «компетентность» стали оперировать лишь в конце 90-х годов XX века, в толковании дефиниций существует разноречие, встречающийся и в научно-педагогической литературе, и в диссертационных исследованиях.

Так, И.А. Зимняя [1] под компетентностью понимает «актуальное, формируемое личностное качество как основывающаяся на знаниях, интеллектуально и личностно обусловленная социально-профессиональная характеристика человека», а компетенция понимается вслед за Н. Хомским, «как совокупность правил (порождения речи), как знание (языка), его структуры». Несколько по-другому разделяет эти понятия А.В. Хуторской: «Компетенция в переводе с латинского *competentia* означает круг вопросов, в которых человек хорошо осведомлен, обладает познаниями и опытом. Компетентный в определенной области человек обладает соответствующими знаниями и способностями, позволяющими ему судить об этой области и эффективно действовать в ней. А.В. Хуторской пытается

разделить данные понятия, имея в виду «под компетенцией некоторое отчужденное, наперед заданное требование к образовательной подготовке ученика, а под компетентностью – уже состоявшееся его личностное качество (характеристику)». Б.И. Хасан считает, что «компетенция – это то, на что претендуют, или то, что назначается, как должное быть достигнутым», а «компетентность – это то, чего достиг из желаемого или вмененного конкретный человек».

Компетенции формируются за счёт педагогических и методологических подходов. При этом используются такие методы, как проектный, творчески-проблемный, исследование ролевых моделей, презентаций идей и др.

Компетенции «закладываются» образовательный процесс посредством технологий, содержания, типа взаимодействия между преподавателями и обучающимися и между самими обучающимися.

Дальнейшее развитие профессионального образования в стране может происходить только на основе компетентностного подхода. Компетенции и результаты образования тесно связаны между собой [3,4].

КРСУ осуществляет эффективные методы обучения современным технологиям программирования. В последние годы существенное внимание уделяется CASE-технологиям разработки ПО (CASE – Computer Aided Software/System Engineering;). Успешное применение этих технологий на практике позволяет в несколько раз увеличить производительность труда программиста, одновременно гарантируя создание корректного, работоспособного и надежного ПО, обладающего свойствами модифицируемости и адаптируемости.

CASE-технология представляет собой совокупность методологий анализа, проектирования, разработки и сопровождения сложных систем ПО, поддержанную комплексом взаимосвязанных средств автоматизации, позволяющих автоматизировать процесс проектирования и разработки ПО. К настоящему моменту дисциплина CASE оформилась в самостоятельное наукоёмкое направление в программотехнике. Существует мнение, что оно является наиболее перспективным и, хотя с этим можно спорить, практически ни один серьёзный зарубежный программный продукт не осуществляется без использования CASE-средств. Известная методология структурного системного анализа SADT (точнее её подмножество IDEF0) принята в качестве стандарта на разработку ПО Министерством обороны США. Более того, среди менеджеров и руководителей компьютерных фирм считается правилом хорошего тона знать основы SADT и при обсуждении каких-либо вопросов нарисовать простейшую диаграмму, поясняющую суть дела.

CASE позволяет создавать не только «правильные» продукты, но и обеспечить «правильный» процесс их создания. Основная цель CASE состоит в том, чтобы отделить проектирование ПО от его кодирования и последующих этапов разработки. Чем больше деятельности будет вынесено в проектирование из кодирования, тем лучше.

CASE-технологии способствуют развитию системного взгляда на разработку ПО. Это обусловлено тем, что разрабатываемая программная система (ПС) фактически является моделью (часто абстрактной) той предметной области, для решения задач или для обслуживания которой ПС создается. Следовательно, как требует практическое применение CASE-технологий, ПС должна обязательно включать, во-первых, такие составляющие (объекты, компоненты), которые свойственны реальной предметной области. Здесь проявляется системная взаимосвязь реального мира и проекта (модели) ПС. Во-вторых, ПС должна содержать вспомогательные составляющие, с помощью которых реализуется обработка, связи, преобразования, наследование, порождение новых составляющих и так

далее. Этот момент требует знания и практического применения системности языка программирования. Другими словами, применение CASE в учебном процессе делает разработку ПО процессом с реальной практической направленностью и обнажает междисциплинарные связи, что придаёт обучению инновационный характер.

Однако процесс преподавания CASE требует решать методические проблемы, связанные с проектированием ПС: априори студенту трудно увидеть те составляющие ПС, которые соответствуют реальной предметной области – они не очевидны, а кругозора, а тем более опыта работы либо не хватает, либо нет совсем. И здесь целесообразно использовать метод решения проблемы «от обратного», т.е. от конца к началу. Дело в том, что любой CASE-программный продукт (CASE-ПП) обладает режимом реверса. Это позволяет трансформировать код уже созданной и работающей ПС в проект, который можно увидеть в тех графических образах – той графической нотации, на которой основан данный CASE-ПП. Таким образом, визуализируется сложный для понимания и умозрительного воспроизведения этап анализа и проектирования ПС.

Например, курсовую работу по разработке ПО, автоматизирующего учёт и контроль товаров на складе, выполняемую на 3-м курсе, можно конвертировать в среде пакета All Fusion (ERwin) в физическую модель данных ERD (Entity Relationship Diagram), в которой отображаются как основные сущности, так и объекты, создаваемые уже для работы программы – элементы интерфейса, вспомогательные объекты, закрывающие концептуальный обзор ПС и её соответствие предметной области.

Затем трансформировать её в логическую модель данных и увидеть реальное соответствие предметной области (рис.1).

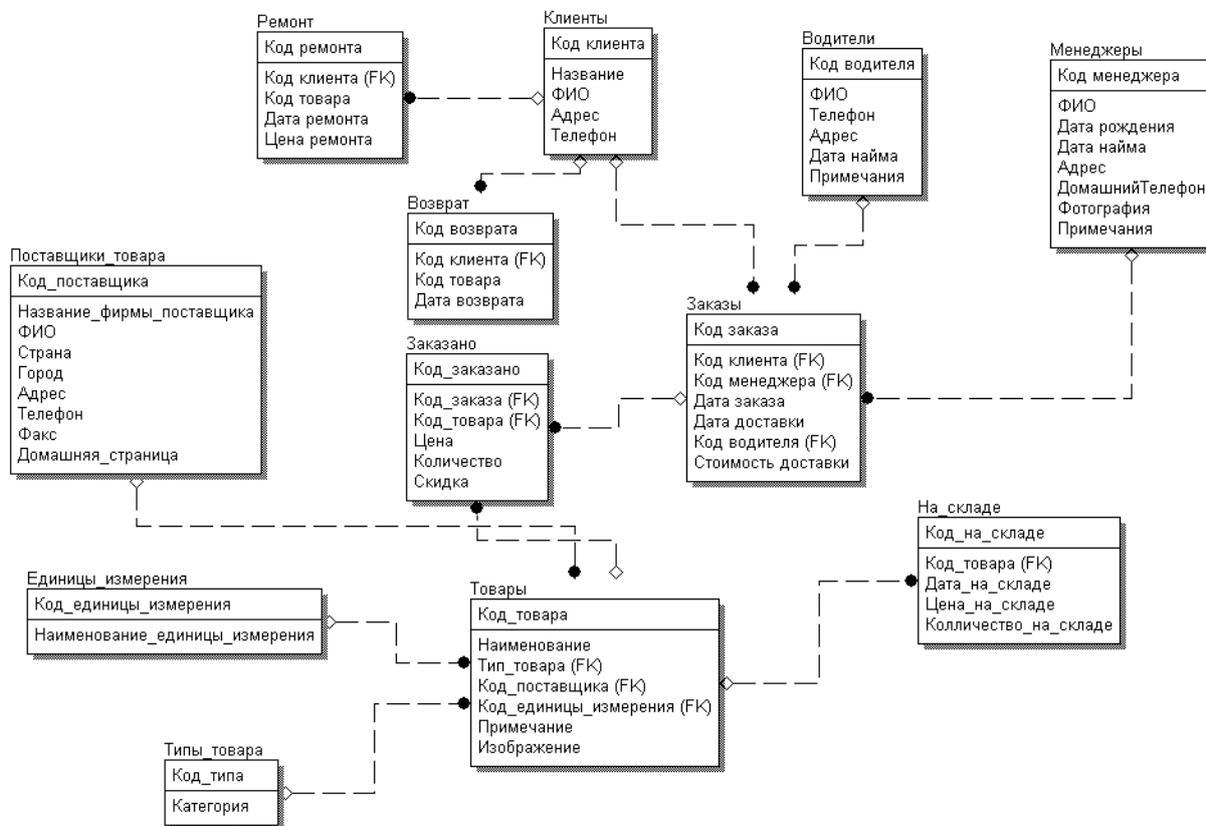


Рис.1. Логическая модель данных, соответствующая ПС «Склад»

Имена сущностей Товары, Поставщики товара, Единицы измерения и др. представляют собой информационные объекты такой предметной области, как «Склад» и составляют её терминологический словарь. Соответствующие атрибуты также напоминают свойства реальных объектов. Между сущностями установлены неидентифицирующие отношения, позволяющие всем им существовать независимо друг от друга вне ПС, что также отображает реальное положение вещей.

Затем модель, представленную на рис.1, можно транспортировать в среду CASE-ПП Power Designer (PD), где визуализируется концептуальная модель, аналогичная логической модели в ERwin, но обладающая своей спецификой. Концептуальная модель должна быть полностью откорректирована в соответствии с нотацией PD:

- должны быть именованы отношения между сущностями и определены типы этих отношений;
- могут быть введены роли для различных сущностей;
- должны быть определены типы данных для всех атрибутов сущностей.

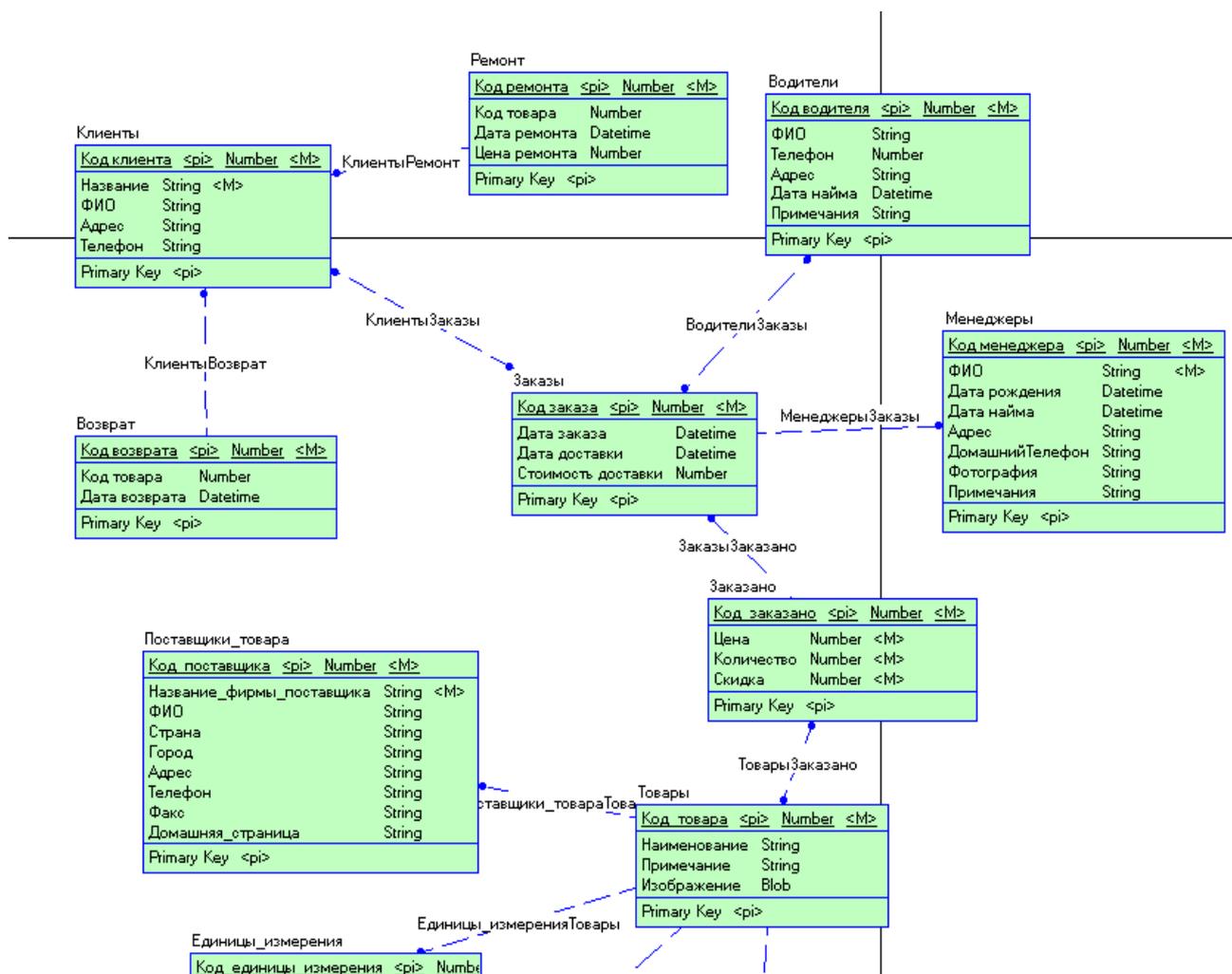


Рис.2. Фрагмент концептуальной модели данных ПС «Склад», выполненный в CASE-ПП Power Designer

На транспортированной модели (рис.2) отношения между сущностями именованы: Клиенты-Заказы, Заказы-Заказано, Менеджеры-Заказы и т.д., что подчёркивает их функциональное значение, а также выведены типы данных для атрибутов.

Корректная концептуальная модель позволяет сгенерировать физическую модель данных в PD (рис.3). На этой модели графическое изображение отношений уже соответствует нотации PD – «ромбик» указывает на основную сущность, а «точка» - на зависимую, типы данных также несколько отличаются от предыдущей модели.

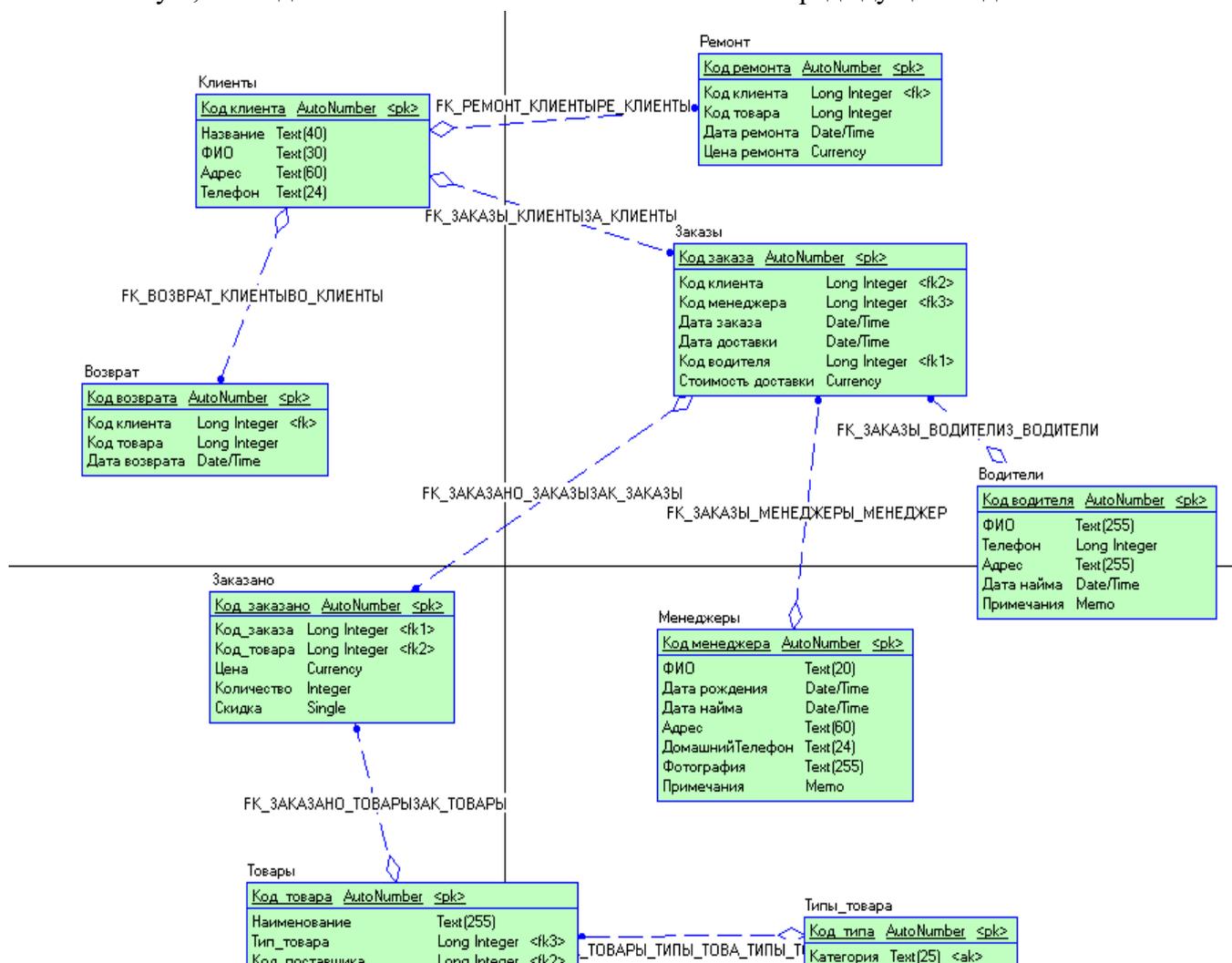


Рис.3. Фрагмент физической модели данных в среде Power Designer

Одной из замечательных особенностей среды разработки PD является возможность конвертирования структурных проектов (моделей) ПО в объектно-ориентированные (ООП) [5,6] и наоборот. Воспользовавшись этим, получаем фрагмент диаграммы классов в нотации языка UML (Unified Modeling Language) – унифицированного языка моделирования систем (любых систем, и программных в том числе), который приводится на рис.4.

Как видно из рис.4, сущности структурной модели становятся в UML-модели классами сущностей, для которых «прописаны» методы, их параметры и параметры атрибутов. Студенты уже знают, что это необходимо делать при написании кода, но здесь они визуально убеждаются, что это же можно делать на этапе проектирования ПО с помощью CASE.

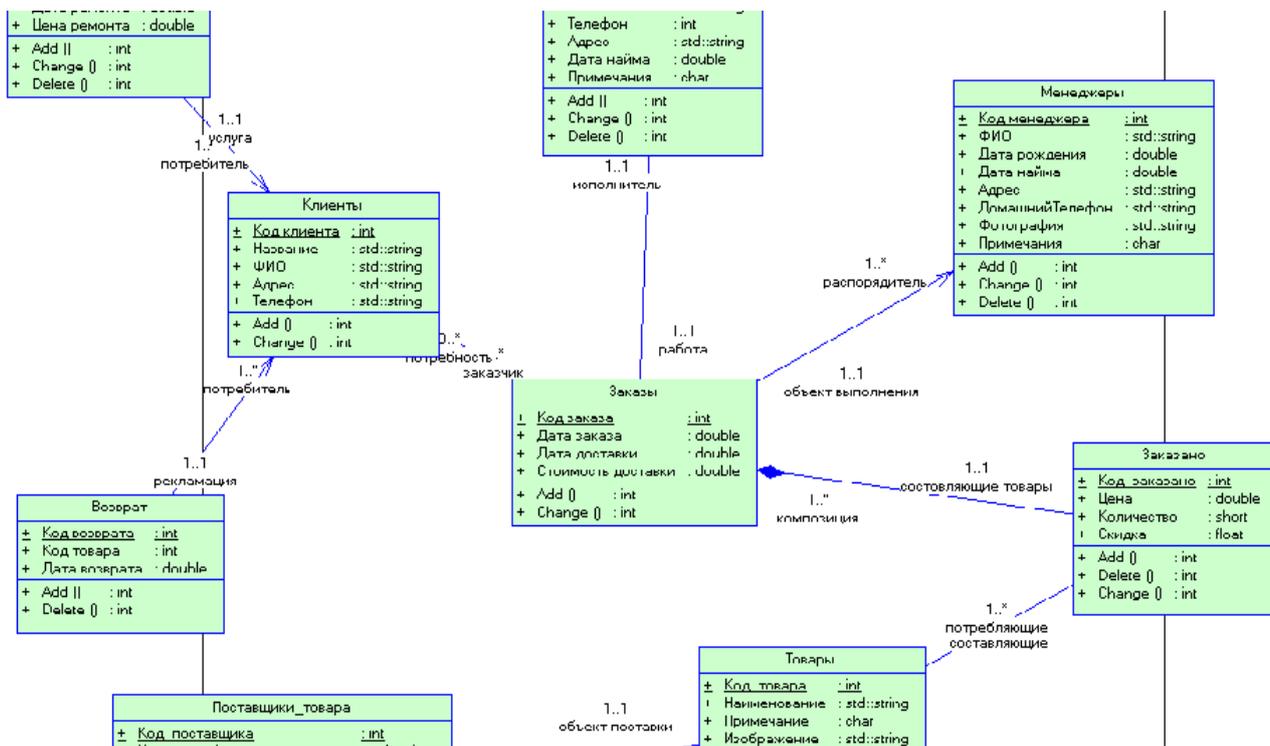


Рис. 4. Фрагмент диаграммы классов языка UML при ООП ПО

Таким образом, практическое использование режима реверса – Reverse Engineering – в CASE-средствах, позволило визуализировать процесс проектирования ПО, его отдельные компоненты (сущности, объекты, атрибуты, связи, методы) а значит, доказать полное соответствие проекта и кода, а также проиллюстрировать этапы ЖЦ [5,6] ПО, хотя и в обратном порядке. Кроме того, студенты визуально наблюдают преемственность двух исторически сложившихся и теоретически обоснованных методов проектирования ПО – структурно-алгоритмического и объектно-ориентированного. Всё это даёт право говорить о реализации таких условий обучения на кафедре ИВТ КРСУ, которые устанавливают требуемый уровень академических и профессиональных компетенций обучаемых, т.к. имеет место визуализированный междисциплинарный, т.е. комплексный, предметно-интегрированный характер подготовки специалистов.

Литература

1. Зимняя И.А. Компетентный подход: каково его место в системе современных подходов к проблеме образования? (Теоретико-методологический аспект) // Высшее образование сегодня. 2006. №8.
2. Болотов В.А., Сериков В.В. Компетентная модель: от идеи к образовательной программе // Педагогика.2003. №10.
3. Концепция модернизации российского образования на период до 2010 года.
4. Татур Ю.Г. Компетентность в структуре качества подготовки специалиста // Высшее образование сегодня. 2004. №3.
5. Вендров А.М. Проектирование программного обеспечения экономических информационных систем: Учебник. – М.: Финансы и статистика, 2006.
6. Калянов Г.Н. CASE-технологии. Консалтинг в автоматизации бизнес-процессов. – М.: Горячая линия – Телеком, 2002.