

Алексеевский Пётр Иванович,

ассистент, кафедра информатики, информационных технологий и методики обучения информатике, Уральский государственный педагогический университет (Екатеринбург); 620075, г. Екатеринбург, ул. К. Либкнехта, 9; e-mail: evilunyu@gmail.com.

**ОБУЧЕНИЕ ПРОГРАММИРОВАНИЮ СТУДЕНТОВ НА ОСНОВЕ МЕТОДОЛОГИИ
УНИФИЦИРОВАННОГО ПРОЦЕССА РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

КЛЮЧЕВЫЕ СЛОВА: обучение программированию; технологии разработки программного обеспечения; объектное моделирование; метод проектов.

АННОТАЦИЯ. Текущее состояние индустрии программного обеспечения обуславливает потребность в ИТ-специалистах, подготовленных к использованию современных технологий разработки программного обеспечения. Для обучения программированию с использованием методологии унифицированного процесса разработана методическая система, устанавливающая связь между фазами унифицированного процесса и этапами обучения.

Alexeevskiy Petr Ivanovich,

Assistant Lecturer of Department of Informatics, Computer Technology and Methods of Teaching Informatics, Ural State Pedagogical University, Ekaterinburg, Russia.

**TEACHING PROGRAMMING TO STUDENTS BASED ON THE METHODOLOGY
OF THE UNIFIED SOFTWARE DEVELOPMENT PROCESS**

KEY WORDS: teaching programming; software development technologies; object modeling; method of projects.

ABSTRACT. The current state of the software industry creates a demand for IT specialists ready to use modern software development technologies. For the purpose of teaching programming based on unified software development process the author uses a special a methodological system which connects its phases with stages of education.

Текущее состояние индустрии программного обеспечения порождает потребность в ИТ-специалистах, подготовленных к использованию современных подходов и технологий разработки программного обеспечения в своей профессиональной деятельности. Процесс подготовки таких специалистов должен включать изучение студентами как алгоритмизации и программирования, так и современных технологических процессов и средств программной инженерии. Такие средства позволяют обеспечить планирование процесса разработки программного проекта, облегчить описание его структурных и поведенческих моделей, а также способствовать быстрому и точному обмену проектной документацией внутри группы разработчиков.

В настоящее время существует множество подходов к организации процесса разработки программного обеспечения. Любая современная организация, ориентированная на разработку ПО, предусматривает определенную последовательность этапов развития программного проекта. И несмотря на то что подробности планирования процесса разработки ПО индивидуальны для каждой такой организации, общая модель технологического процесса подчиняется строгой закономерности.

Одной из наиболее распространенных методологий разработки ПО является «Уни-

фицированный процесс разработки программного обеспечения» («The Unified Software Development Process») [5; 2]. Основными приоритетами этой методологии являются так называемые «варианты использования» разрабатываемого программного продукта (в том числе планируемый функционал и интерфейсы), его архитектура, а также экономические факторы, связанные с процессом разработки и внедрения. Эффективность методологии подтверждена многолетним опытом ее реализации в различных организациях, среди которых можно выделить такие крупные компании, как IBM, «Rational Software», «Oracle», «Eclipse Foundation» и др.

При всем многообразии путей реализации унифицированного процесса основа методологии остается неизменной и включает в себя четыре базовые фазы: «Начальная стадия», «Уточнение», «Построение» и «Внедрение».

Другой особенностью методологии унифицированного процесса является использование графического метапредметного языка объектного моделирования, получившего название UML (unified modeling language, унифицированный язык моделирования). Средства этого языка дают возможность представить структурную и поведенческую модель проекта в наглядной форме, что, в совокупности со строгим опи-

санием компонентов и их отношений, позволяет использовать его в качестве инструмента для взаимодействия участников процесса разработки. Кроме того, метапредметность языка UML может быть реализована и в других областях, не связанных с разработкой ПО.

Данные особенности методологии унифицированного процесса разработки программного обеспечения привели к росту ее популярности и принятию во множестве организаций, ориентированных на разработку и сопровождение программного обеспечения. Распространенность методологии позволяет сделать вывод о целесообразности обучения ИТ-специалистов работе в условиях реализации унифицированного процесса.

Для полноценного обучения программированию при реализации методологии унифицированного процесса необходимо представить теоретический материал в соответствии с последовательностью этапов разработки программного проекта, закрепляя его изучение практическими заданиями. Ввиду того что разработка крупных программных проектов осуществляется группами разработчиков, требуется предусмотреть возможность построения практических заданий с учетом коллективного характера деятельности. Реализация данного подхода в Уральском государственном педагогическом университете потребовала создания методической системы, учитывающей технологические особенности унифицированного процесса разработки ПО, такие как подготовка проектной документации, формирование подгрупп разработчиков, применение средств управления версиями и др. Апробация методической системы обучения осуществлялась в течение пяти лет в рамках специализации «Компьютерные игровые технологии в образовании» при подготовке студентов по специальности «Информатика», а также при обучении программированию студентов других специальностей.

В основу методической системы легли следующие положения:

- использование методологии унифицированного процесса на протяжении всего курса обучения программированию;
- реализация метода проектов;
- организация практических заданий для небольших (3–4 человека) групп студентов;
- выбор метапредметного направления разработки программного проекта.

Таким образом, была разработана методическая система, в которой порядок изучения программирования основан на технологическом процессе, используемом при разработке крупных программных проектов, включающих в себя функциональные компоненты различного назначения.

В качестве предметной области для практической составляющей методической системы были выбраны компьютерные игры. Данный выбор обусловлен тем, что игровые программы представляют собой один из наиболее сложных в разработке классов программного обеспечения, обладающий всеми необходимыми характеристиками для использования в качестве практического компонента (метапредметность, архитектурная сложность, коллективный характер разработки). При организации практических заданий обучаемые делятся на небольшие группы по 3–4 человека, осуществляющие работу над общим в пределах группы проектом на протяжении всего курса обучения.

Для реализации подходов унифицированного процесса разработанная методическая система предполагает разбиение процесса обучения на несколько этапов. Количество и содержание этапов приводятся в соответствии с фазами унифицированного процесса, таким образом выделяется четыре этапа обучения. Порядок этапов и их соответствие фазам унифицированного процесса приведены на рисунке.

Методическая система предусматривает изучение курса в порядке, соответствующем порядку следования фаз унифицированного процесса, допуская при этом возможность возврата на предыдущий этап (с определенными ограничениями).

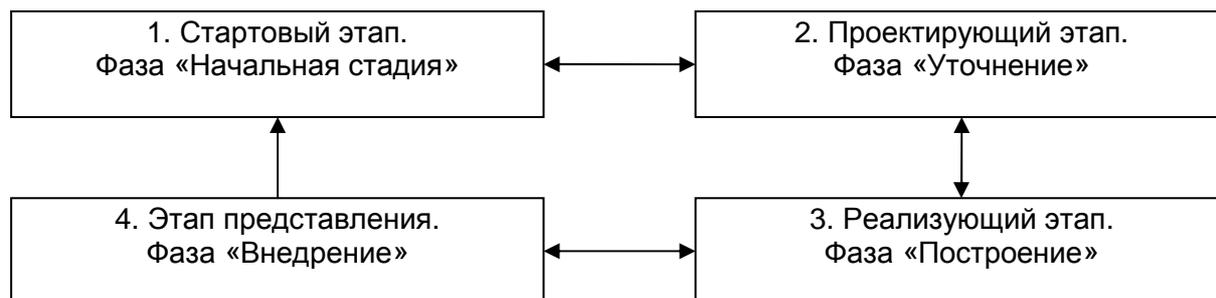


Рис. Этапы обучения и их соответствие фазам унифицированного процесса

Для каждого из этапов обучения определены следующие параметры:

- 1) входные требования, предъявляемые к уровню знаний и умений обучаемого;
- 2) цель текущего этапа;
- 3) содержание обучения;
- 4) методы и формы обучения;
- 5) формы и содержание контроля;
- 6) результат этапа.

На *стартовом этапе* специальные входные требования не предусмотрены. Целью данного этапа является формирование у будущего ИТ-специалиста знаний и умений в области проектирования.

Содержание обучения на данном этапе включает в себя изучение структуры и особенностей унифицированного процесса, его основных ориентиров и средств. Формируются группы студентов, которые будут заниматься в дальнейшем разработкой и реализацией программного проекта, общего для группы. Осуществляется выбор тематики разрабатываемой программы и формулирование ее основных характеристик. На данном этапе рассматриваются базовые принципы формирования архитектуры программы и процесс планирования разработки программного проекта.

Основным методом обучения на данном этапе являются проблемные лекции.

Для оценки результатов обучения используются формы текущего индивидуального контроля. Содержание контроля – знания в области унифицированного процесса разработки и проектирования программного обеспечения.

Результатом обучения на данном этапе является сформированность знаний в области организации унифицированного процесса разработки программного обеспечения.

После достижения этого результата осуществляется переход к *проектирующей* *этапу*, цель которого – формирование знаний и умений в области объектного моделирования.

На проектирующем этапе изучаются основы объектного моделирования с использованием языка UML, а также основные принципы объектно ориентированного программирования без учета специфики языков программирования. Рассматриваются структурные и поведенческие модели программного обеспечения, основные алгоритмы взаимодействия программных модулей. Уточняется тематика разрабатываемого группами студентов программного проекта и осуществляется дальнейшее планирование его разработки.

Методы обучения, используемые на данном этапе, включают в себя проблемные лекции, имитационные упражнения, игровое проектирование, а также ситуационные методы.

Оценка результатов обучения осуществляется методами тематического текущего контроля в индивидуальной и групповой форме. Оцениваются результаты разработки архитектуры программных проектов в каждой из групп студентов.

Результатом обучения является сформированность знаний и умений в области объектного моделирования и языка UML.

Целью реализующего этапа является формирование знаний и умений в области программирования на языках высокого уровня.

На этом этапе осуществляется изучение языков программирования высокого уровня C и C++, а также особенностей реализации различных алгоритмов на этих языках. Особое внимание уделяется возможности использования сторонних библиотек, позволяющих осуществлять работу с графическими и звуковыми возможностями программно-аппаратной платформы, а также реализующих наиболее часто используемые алгоритмы обработки данных. Рассматриваются методы оценки сложности алгоритмов, способы оптимизации программного кода на различных уровнях, средства отладки и профилирования. Осуществляется реализация функциональных компонентов программных проектов, разрабатываемых студентами. Для взаимодействия студентов внутри группы используются технологии управления версиями исходного кода программы, поддерживающие коллективную работу над проектом, такие как «Git» и «Subversion».

Для практической реализации компонентов программы осуществляется распределение модулей и подсистем между студентами внутри группы.

Основные методы обучения на данном этапе включают в себя проблемные лекции, имитационные упражнения, ситуационные методы, тематические дискуссии.

Для оценки результатов обучения применяются формы группового и индивидуального текущего контроля. Проверяются результаты реализации компонентов программного проекта в каждой из групп студентов.

На данном этапе результатами обучения являются сформированность знаний и умений в области программирования на языках C и C++, обретение опыта коллективной разработки программ, а также рабочие версии программ, разработанных в рамках коллективных проектов.

Этап представления предполагает формирование знаний и умений в области разработки программной документации. На этом этапе также осуществляется публичная защита разработанных программных продуктов.

Основными методами обучения на данном этапе являются проблемные лекции, групповые консультации и имитационные упражнения.

Содержание включает в себя изучение средств генерации программной документации по объектной модели и исходному коду программы. Рассматриваются автоматические генераторы документации на основе внедренных в исходный код комментариев определенного вида, такие как «Doxugen», а также средства и технологии ручного создания документации, например, для встроенных систем подсказки.

Осуществляется индивидуальный тематический итоговый контроль; групповой контроль осуществляется путем публичной защиты разработанного программного продукта и его апробации методом экспертных оценок. Допускается также публикация разработанных материалов в студенческих сборниках статей.

Результатами данного этапа являются сформированность знаний и умений в области создания программной документации и обретение опыта коллективной разработки, реализации и внедрения программного обеспечения.

Особенностью данной методической системы является ее пригодность как для обучения студентов программированию,

так и для повышения квалификации ИТ-специалистов, чья профессиональная деятельность связана с разработкой, реализацией и внедрением программного обеспечения. Основа методической системы допускает адаптацию содержания обучения к разработке различных классов программных продуктов, включая системы имитационного моделирования и симуляции, системы управления производственными процессами, сложные веб-приложения и порталы и т. д. Поскольку в структуре содержания методической системы делается акцент на технологиях разработки программного обеспечения, а не на фиксированных программных продуктах, их реализующих, набор используемого при обучении инструментального программного обеспечения может варьироваться в зависимости от состояния рынка и возможностей реализующего данную методическую систему вуза.

Педагогический эксперимент по внедрению данной методической системы в процесс обучения студентов Института информатики и информационных технологий Уральского государственного педагогического университета позволил выявить применимость и целесообразность методической системы для обучения программированию студентов как технических, так и педагогических специальностей.

ЛИТЕРАТУРА

1. Алексеевский П. И. Изучение средств управления версиями в рамках курса по программированию // Инновационные технологии в образовательном процессе высшей школы / Урал. гос. пед. ун-т. Екатеринбург, 2012.
2. Алексеевский П. И., Лапенков М. В. Выбор программного обеспечения для проведения практических занятий по программированию на С и С++ // Информатика и образование. 2010. № 4. С. 48–52.
3. Зарукина Е. В., Логинова Н. А., Новик М. М. Активные методы обучения: рекомендации по разработке и применению : учеб.-метод. пособие. СПб. : СПбГИЭУ, 2010.
4. Alexeevskiy P. I. Teaching Computer Programming to Students of Pedagogical Specialties through the Process of Game Development // The Ethos of the Academe: Standing the Test of Time / Ariel Univ. Ariel, 2013.
5. Jacobson I., Booch G., Rumbaugh J. The Unified Software Development Process. Addison Wesley Longman, 1999.

Статью рекомендует канд. техн. наук, доц. М. В. Лапенков.