

онные процессы, математические основы информатики, алгоритмы и исполнители, коммуникационные технологии. Ценность этого метода обучения обусловлена тем, что по характеру вопросов учитель может судить о глубине знаний учащихся, степени их познавательной активности, стремлении понять сущность рассматриваемых явлений, процессов, т. е. вопросы учащихся являются своеобразным средством «обратной связи». Кроме того, зачастую вопрос учащегося включает в цепочку последовательно связанные между собой другие вопросы: решение первого порождает второй, третий, и таким образом учащиеся вовлекаются в активную работу. Вопросы в этом случае становятся не только критерием глубины знаний и интереса учащихся, но и средством, поддерживающим этот интерес.

Мы полагаем, что применение методов эвристической беседы и дидактической игры способствует росту интеллектуальной активности учащихся на уроке, развитию мышления, глубокому пониманию учащимися изучаемого материала, умению применить имеющиеся знания для решения новых познавательных и практических задач, а следовательно, формированию и развитию познавательных умений учащихся.

ЛИТЕРАТУРА

1. Механизмы формирования универсальных учебных действий на основе дидактической системы деятельностного метода обучения «школа 2000...». [Электронный ресурс]. Режим доступа: http://ito.edu.ru/sp/SP/SP-0-2011_05_10.html

2. Работа педагогов по формированию УУД. [Электронный ресурс]. Режим доступа: http://www.gym9.ru/gymnasium/rubric_311/rubric_313/

3. Формирование универсальных учебных действий в основной школе. От действия к мысли. Система заданий. Автор: А. Г. Асмолов, Г. В. Бурменская, И. А. Володарская, О. А. Карабанова, Н. Г. Салмина, С. В. Молчанов Редактор: Александр Асмолов Издательство: Просвещение Страниц 160 стр.

4. Эвристическая беседа. [Электронный ресурс]. Режим доступа: <http://osvarke.info/229-yevristicheskaya-beseda.html>

ПОДХОД К РАЗРАБОТКЕ ПРОГРАММ ЧЕРЕЗ МОДУЛЬНОЕ ТЕСТИРОВАНИЕ

М.О.Морозов

Научный руководитель: Д.А. Емельянов, к.т.н., доцент
ФГБОУ ВПО «УрГПУ», Екатеринбург, Россия

Обучение программированию в школе в рамках предмета «информатика», а также преподавание языков программирования в вузах должно способствовать подготовке некоторого процента будущих кадров, способных заниматься профессиональной разработкой программного обеспечения.

Но учащиеся, в дальнейшем выбравшие данную специальность, непременно столкнутся с проблемой недостатка знаний и необходимости заниматься самообразованием. Это вполне естественно и так можно сказать про многие быстро развивающиеся технические области знаний. И только людям, действительно занимающимся разработкой программного обеспечения и одновременно знакомым с учебной программой школ и вузов, видны наиболее критические моменты разрыва между тем, что требуется в реальной работе и тем, чему обучают.

Лично для меня на текущий момент наиболее существенным пробелом в образовательной программе школ и вузов является игнорирование следующей темы: «Подход к разработке программ через модульное тестирование».

Модульное тестирование, или юнит-тестирование (англ. unit testing) — процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы.

Идея состоит в том, чтобы писать тесты для каждой нетривиальной функции или метода. Это позволяет достаточно быстро проверить, не привело ли очередное изменение кода к регрессии, то есть к появлению ошибок в уже оттестированных местах программы, а также облегчает обнаружение и устранение таких ошибок.

Цель модульного тестирования — изолировать отдельные части программы и показать, что по отдельности эти части работоспособны.

Этот тип тестирования обычно выполняется программистами.

Поощрение изменений. Модульное тестирование позже позволяет программистам проводить рефакторинг, будучи уверенными, что модуль по-прежнему работает корректно (регрессионное тестирование). Это поощряет программистов к изменениям кода, поскольку достаточно легко проверить, что код работает и после изменений.

Упрощение интеграции. Модульное тестирование помогает устранить сомнения по поводу отдельных модулей и может быть использовано для подхода к тестированию «снизу вверх»: сначала тестируются отдельные части программы, затем программа в целом.

Документирование кода. Модульные тесты можно рассматривать как «живой документ» для тестируемого класса. Клиенты, которые не знают, как использовать данный класс, могут использовать юнит-тест в качестве примера.

Отделение интерфейса от реализации. Поскольку некоторые классы могут использовать другие классы, тестирование отдельного класса часто распространяется на связанные с ним. Например, класс пользуется базой данных; в ходе написания теста программист обнаруживает, что тесту приходится взаимодействовать с базой. Это ошибка, поскольку тест не должен выходить за границу класса. В результате разработчик абстрагируется от соединения с базой данных и реализует этот интерфейс, используя свой собственный mock-объект. Это приводит к менее связанному коду, минимизируя зависимости в системе.

Модульное тестирование в образовании

Итак, мы теперь имеем представление об этом. Но действительно ли этому следует обучать? Можно смело заявлять, что за последние годы данный подход разработки ПО стал основополагающим. Если лет 5 назад, многие еще присматривались к нему, опасаясь, что это очередное проходящее новомодное веяние. То сейчас эффективность данного подхода уже многократно доказана. В большей части современных open-source репозиториях можно встретить не только папку «src», но и папку «test». Если в проекте объем тестового кода равен продакшн коду или даже превышает его, то это ни у кого уже не вызывает удивления.

Итак, допустим, мы определились и решили, что нужно начинать обучать студентов и школьников модульному тестированию. Но программа же не резиновая. Может от чего-то можно отказаться? Да, можно значительно сократить ту часть, посвященную вопросу рантайм отладки. Так как эти два подхода взаимозаменяемы. Грубо говоря, можно считать, что рантайм отладка это устаревшая методика выявления ошибок, а модульное тестирование современная. На самом деле лучше уметь и то и то, так как эти два подхода хорошо дополняют друг друга.

Остаётся и самый важный вопрос: «Как вести преподавание этой темы?» Легче всего, конечно, было бы обучать модульному тестированию на основе языка программирования Java, но поскольку данный навык лучше начинать прививать со школьных лет, то придется остановиться на Delphi, так как сейчас он является самым распространённым языком программирования, преподаваемым в школах.

Фреймворк модульного тестирования для Delphi носит название DUnit. Предполагается, что практическая работа, будет вестись именно с ним. В идеале, умение работать с данным инструментом ученикам нужно привить как можно на более раннем этапе, потому что даже изучение самого языка Pascal, можно вести не через традиционную консоль и input/output файлы, а через написание тестов.

Как ведётся обучение в традиционном виде в школе

1. Преподаватель даёт новый теоретический материал по возможностям языка Pascal.

2. Дается практическая задача на отработку эти новых возможностей.
3. Формулируются входные и выходные данные, которые нужно вести и получить. Через консоль или чрез input/output файлы.
4. Ученик пишет программу.
5. Постоянно перезапускает ее и переделывает, пытаясь получить правильные выходные данные, путем устранения ошибок в программе.
6. В сложных ситуациях применяется рантайм отладка.

Как предполагается вести обучение для людей, заведомо знакомых с модульным тестированием

1. Преподаватель дает новый теоретический материал по возможностям языка Pascal.
2. Дается практическая задача на отработку эти новых возможностей.
3. Ученик вначале пишет юнит тесты, которые утверждают, какой функционал необходимо разработать.
4. Ученик пишет необходимый функционал.
5. Тесты автоматически перезапускаются и проверяют написанный функционал до тех по, пока не будут устранены все ошибки.

В отличие от input/output подхода, подход через тесты позволяет написать программу, которая состоит из множества отдельных подзадач, и все эти отдельные подзадачи также легко независимо протестировать.

Практическая часть: Delphi + DUnit

Может сложиться ощущение, что данный подход позволяет тестировать только программы, функционалом которых является различные вычисления и преобразования данных. Но на самом деле это не так. Модульное тестирование отлично подходит и для разработки GUI и для визуальных анимации и игр. В практической части моего доклада будет рассмотрено применение модульного тестирования для написания двух совершенно различных приложений.

ЛИТЕРАТУРА

1. Приемы объектно-ориентированного проектирования. Паттерны проектирования. / Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж., СПб.: Питер, 2010. 366 с.
2. Кериевски Д. Рефакторинг с использованием шаблонов. М.: И.Д. Вильямс, 2008. 400 с.
3. Ларман К. Применение UML 2.0 и шаблонов проектирования. Практической руководство. 3-е изд. М.: И.Д. Вильямс, 2009. 736 с.
4. Макконнелл С. Совершенный Код. СПб.: Питер, 2008. 896 с.

5. Фаулер М. Архитектура корпоративных программных приложений. М.: И.Д. Вильямс, 2008. 544 с.
6. Фаулер М. Рефакторинг: улучшение существующего кода. СПб.: Символ-Плюс, 2009. 432 с.
7. Физерс М. Эффективная работа с унаследованным кодом. М.: И.Д. Вильямс, 2009. 400 с.
8. Эккель Б. Философия Java. Библиотека программиста. 4-е изд. СПб.: Питер, 2010. 640 с.

ИСПОЛЬЗОВАНИЕ ГРУППОВЫХ ПРОЕКТОВ В УЧЕБНОМ ПРОЦЕССЕ

В.Ю. Пирогов, зав. кафедрой прикладной информатики,
электронная почта: pirogov@shadrinsk.net
ФГБОУ «ШГПИ», Шадринск, Россия

Аннотация

В статье рассматриваются различные аспекты создания проектов информационных систем в контексте процесса обучения. В статье содержится анализ моделей групповых проектов, характеристики этих моделей и некоторый опыт автора по использованию этой формы учебной работы.

Ключевые слова: *информационные системы, базы данных, программное обеспечение, жизненный цикл, дипломный проект, проектная группа, рабочая группа, промежуточный контроль.*

Abstract

This paper discusses different aspects of projection of information systems in the context of the learning process. The paper contains analysis of models of group projects, the characteristics of these models and some experience of the author in using of this form of teaching work.

Keywords: *information system, database, software application, life cycle, degree project, group project, working group, intermediate control.*

Общие замечания

Под термином «проект» в данной статье будем понимать целенаправленную деятельность по созданию программного обеспечения (далее ПО) или, в более узком смысле, информационной системы (далее ИС). Проект является важнейшим элементом учебного процесса по инженерным специальностям, так или иначе связанным с программированием и базами данных (далее БД). Именно в проекте студент на практике может понять особенности реальной разработки ПО, те сложности и проблемы, о которых говорится на лекциях, но которые могут быть усвоены только при создании реального (пусть в значительной степени и учебного, т.е. облегченного) продукта. Примером может служить в частности влияние структуры БД на создаваемое ПО. Особенности этого влияния, первичность структуры БД по отношению к ПО, хорошо усваи-