

Одна з задач, яка ставиться перед системою підтримки прийняття рішень в даній роботі, полягає в аналізі вихідного коду програмного продукту, виявленні найбільш слабких місць та формуванні практичних рекомендацій щодо рефакторингу.

Рефакторинг коду це процес зміни початкового коду програми, що не змінює її поведінки і ставить метою полегшити розуміння її роботи та/або спростити її подальшу підтримку. Зазвичай рефакторинг будь-якої програми це внесення невеликих змін в її код, кожне з яких не змінює саму поведінку програми, але якимось спрощує і/або поліпшує код.

Дуже важливо, щоб рефакторинг виконувався саме невеликими частинами, тому що коли змінюється невелика частина коду - значно простіше простежити за правильністю змін і не наробити помилок. Якщо ж змінюються відразу великі ділянки коду, то є дуже велика ймовірність наробити помилок, в результаті яких програма може взагалі перестати працювати.

Основна мета рефакторингу це зробити код простіше і зрозуміліше. Якщо після переробки код не став краще і зрозуміліше - значить рефакторинг не робився, або він не вдався. При цьому не можна плутати поняття рефакторингу з оптимізацією. У результаті оптимізації код стає швидше, але зовсім не обов'язково простіше і зрозуміліше, рефакторинг ж служить саме для спрощення і поліпшення зрозумілості коду.

Нижче сформульовані основні правила, ґрунтуючись на яких, можна зрозуміти, що код вимагає переробки:

- Якщо у програмі є дублювання коду, то майже напевно потрібно проводити рефакторинг. Повторюваний код у програмі - основне джерело помилок. Якщо у програмі якась дія виконується в декількох різних місцях, але одним і тим же кодом - просто необхідно винести цей код в окрему функцію і викликати її. Інакше висока ймовірність того, що одного разу буде змінений код в одному місці, але буде не виправлений аналогічний код в іншому місці і виникнуть помилки.

- У вашій програмі є дуже довгі методи/функції. Як правило, людина не може повністю сприймати і оцінювати правильність коду, якщо цей код займає більше 2-3 десятків рядків. Такі методи і функції слід розділяти на декілька більш дрібних і робити одну спільну функцію, яка буде послідовно викликати ці методи.

- Довгий список параметрів функції/методу/конструктора. Велика кількість параметрів зазвичай не тільки ускладнює розуміння того, що робить цей метод або функція, але і ускладнює розуміння коду, що використовує ці функції. Якщо необхідно, щоб функція приймала дуже багато параметрів - просто винесіть ці параметри в окрему структуру (або клас), давши цій структурі розумне і зрозуміле ім'я, і передавайте функції посилання (або покажчик) на об'єкт цієї структури або класу.

- Великі класи так само вимагають рефакторингу. Якщо в програмі є один або кілька великих (більше пари-тришки сотень рядків коду) класів, слід нетайно розділити їх на більш дрібні і включити об'єкти цих класів в один загальний клас. Причина цього та ж сама, що і в попередньому пункті.

- Занадто багато тимчасових змінних так само є ознакою поганого коду, який вимагає рефакторингу. Як правило, багато тимчасових змінних зустрічаються в надмірно "роздутих" функціях - коли робиться рефакторинг таких функцій, швидше за все і кількість тимчасових змінних в кожній з них стане менше і код стане значно зрозуміліше і зручніше.

- Багато "безладно" збережених даних, які пов'язані логічно і їх можна було б об'єднати в структуру, або клас. Логічно пов'язані дані завжди варто зберігати в структурах/класах, навіть якщо це всього 2-3 змінних.

Результатом проведеної роботи є СППР, яка синтаксично аналізує код, надає розрахунки метрик вихідного коду та видає список порад щодо рефакторингу.

Література:

1. Фаулер М. Рефакторинг. Улучшение существующего кода. / М. Фаулер, К. Бек, Д. Брант, У. Андайт, Д. Робертс, Е. Гамма // *Издательство «Символ-Плюс»*, 2008. – 432с.
2. Фаулер М. Чистый код. / М. Фаулер // *Издательство «Питер»*, 2010. – 464с.
3. Гамма Е. Приемы объектно-ориентированного проектирования. Паттерны проектирования. / Е. Гамма, Р. Хелм, Р. Джонсон, Д. Виллисидес // *Издательство «Питер»*, 2007. – 366с.
4. Lorenz M. *Object-Oriented Software Metrics* // M. Lorenz, J. Kidd // «Prentice Hall», 1994. - 146pp.

СОЗДАНИЕ ИНТЕРФЕЙСА ВЗАИМОДЕЙСТВИЯ СИСТЕМЫ MOODLE И ПРОГРАММЫ ТЕСТИРОВАНИЯ МОРЯКОВ

Каминская Н.Г., Кравцова Л.В.

Украина, г. Херсон,

Херсонская государственная морская академия

Аннотация. *Рассмотрен вопрос модификации модульной объектно-ориентированной динамической учебной среды Moodle с целью создания интерфейса взаимодействия с программами тестирования моряков.*

Постановка проблемы. Дистанционная форма обучения на сегодняшний день является наиболее активно внедряемым направлением в образовании. Интерактивное взаимодействие преподавателей со студентами с использованием информационно-коммуникационных технологий находит широкое применение. Одним из первых вопросов, возникающих в процессе создания курса дистанционного обучения, является вопрос выбора платформы, на которой будет реализован дистанционный курс. Херсонская государственная морская академия (ХГМА), как и большинство других учебных заведений Украины, использует инструментальную среду построения образовательного контента Moodle. Кроме того, ХГМА имеет мощную тренажерную базу и большое количество специализированных обучающих и тестирующих программ подготовки моряков. При этом возникает проблема интеграции результатов обучения в системе Moodle и в специальных обучающих программах в единую базу данных в Moodle. В данной статье предлагается метод решения этой проблемы.

Анализ последних исследований и публикаций. Дистанционная форма обучения наиболее соответствует современному уровню развития общества. На сегодняшний день в мире накоплен достаточно большой опыт внедрения

систем дистанційного навчання в процесі підготовки спеціалістів різних профілей [1,2]. Вибір платформи для реалізації цієї цілі во многом визначає дальнішу стратегію навчального закладу в напрямленні розвитку дистанційного навчання (ДОН) [3,4,5].

Система дистанційного навчання (СДОН) – це програмне забезпечення для організації дистанційної форми навчання, додаткової системи підтримки навчального процесу, електронного документообігу, проведення консультацій, створення електронних навчальних матеріалів, адміністрування і оцінки успішності в рамках вивчаємої дисципліни. Вивчення досвіду впровадження СДОН іншими висшніми навчальними закладами допомагає значно скоротити шлях від ідеї впровадження ДОН до її реалізації [4, 5].

Ціль досліджень – створення інтерфейсу взаємодії системи Moodle і програмного комплексу тестування моряків.

Основна частина. Сучасне дистанційне навчання будується на використанні наступних основних елементів: середовища передачі інформації (наприклад, інформаційні комунікаційні мережі) і методів, залежних від технічного середовища обміну інформацією.

Одним з перших питань, що виникають в процесі створення курсу дистанційного навчання, є вибір платформи, на якій буде реалізований дистанційний курс. Для того щоб прийняти в цьому виборі правильне рішення, яке, можливо, вплине на долю всього проекту, необхідно проаналізувати можливості відомих систем управління навчанням.

Вибір платформи залежить від багатьох причин. Зокрема, безумовно, привабливими є вільно розповсюджені системи, які відповідають вимогам відповідності стандартам інформаційних навчальних систем, забезпечення зручності освоєння і використання преподавателями навчальних закладів. Всім цим і багатьом іншим необхідним якостям відповідає система побудови освітнього контенту Moodle (Modular Object-Oriented Dynamic Learning Environment) [6,7]. Даний програмний продукт розроблений відповідно до стандартам інформаційних навчальних систем, це вільно розповсюджуєма система побудови освітнього контенту, що дозволяє паралельно вивчати інструментарій ДОН Moodle і створювати свій навчальний курс.

Moodle (модульна об'єктно-орієнтована динамічна навчальна середовище) поєднує в собі кілька класів систем:

- система управління сайтом (CMS);
- система управління навчанням (LMS);
- віртуальна середовище навчання (VLE).

Moodle розповсюджується вільно, як програмне забезпечення з відкритим вихідним кодом (в відповідності з GNU Public License). Система Moodle пройшла сертифікацію відповідності стандарту SCORM 1.2. Moodle – це система управління контентом сайту (Content Management System - CMS), спеціально розроблена для створення онлайн-курсів преподавателями. Такі e-learning системи часто називаються системами управління навчанням (Learning Management Systems - LMS) або віртуальними освітніми середовищами (Virtual Learning Environments - VLE).

Moodle – це інструментальна середовище для розробки як окремих онлайн-курсів, так і освітніх веб-сайтів. Відкритий програмний код Moodle дозволяє вносити зміни, всілякі модифікації, які робляться практично щодня спеціалістами світового співтовариства в області розробки програмного забезпечення для підтримки освіти.

Автор Moodle Мартін Дугіамас пропонує широкий спектр можливостей для повноцінної підтримки процесу навчання в дистанційній середовищі – різноманітні способи представлення навчального матеріалу, перевірки знань і контролю успішності.

В Moodle передбачено 15 типів інтерактивних навчальних матеріалів. Кожен з цих типів має багато налаштувань і представлень. Навчальний матеріал можна підготувати також і в інших програмах.

Всі матеріали мають своєю метою підвищення якості знань навчаних. Одні типи матеріалів забезпечують взаємодію студента і преподавателя, інші – взаємодію студентів між собою. Крім перерахованих, Moodle містить також багато додаткових плагінів і програмних модулів.

В даний момент систему Moodle використовують для навчання найбільші університети світу. Херсонська державна морська академія, як і абсолютна більшість навчальних закладів України і світу, також використовує інструментальну середовище побудови освітнього контенту Moodle з метою підвищення якості навчання і впровадження його дистанційної форми. Оскільки Moodle є інструментальною середовищем з відкритим програмним кодом, це дозволяє вносити всілякі зміни і модифікації на усмотрение навчального закладу, який використовує цю середовище.

Серверний центр дистанційного навчання академії забезпечує весь цикл створення і викладання дистанційного курсу.

Херсонська морська академія має потужну тренажерну базу для відпрацювання практичних навичок курсантів в навчальному процесі. Наявність тренажерного комплексу і лабораторій в ХГМА дає можливість курсантам вивчити питання, пов'язані з професійною підготовкою, і удосконалити практичні навички відповідно до вибраними спеціальностями. Така підготовка необхідна перед проходженням практики в реальних умовах, що в значній мірі підвищує конкурентоспроможність курсантів і випускників Херсонської державної морської академії. Відпрацювання навичок на тренажерах проходить відповідно до програмою підготовки курсантів. Програма передбачає також поточний і підсумковий контроль. А це означає, що результати контролю повинні бути внесені в загальну базу даних по кожному курсанту, незалежно від того, яку форму навчання він вибрав – стаціонар, заочну, дистанційну. Крім цього, всі курсанти старших курсів проходять комп'ютерне тестування. Зокрема, це програми, які використовуються морськими керівними компаніями для тестування моряків при пристроєнні на роботу. Очевидно, результати тестування кожного курсанта, як і результати проходження підготовки на тренажерній базі, повинні автоматично фіксуватися в базі даних системи Moodle, поповняючи його облікову записку і беручи участь в підсумковій атестації нарівні з оцінками по дисциплінам.

В цьому і полягає основна проблема.

Плановое компьютерное тестирование по специальным программам проводится в компьютерных аудиториях академии. Программа тестирования установлена на каждом компьютере. Одна из таких программ – CES (Competence Evaluation System) – компьютерная программа комплексной проверки знаний моряков на соответствие требованиям международной конвенции STCW-78/95. CES (SEAGULL TEST) — это норвежская программа для обучения и тестирования моряков, система оценки компетентности, которая содержит более 5000 вопросов в семи областях STCW '95. Данный тест обязателен к прохождению всех моряков (рядовых и офицеров), которые становятся на учет в компании V.Ships впервые.

Тест считается пройденным в случае если:

- Общий балл 70 % и более (не менее 60% по каждой категории теста).
- Для Navigation в разделе «Detailed test» — 80% и более.

При прохождении тестирования курсант заполняет электронную анкету, после чего открывается доступ к программе. Время тестирования программой не ограничено. По окончании тестирования выдается электронный лист, на котором распечатаны данные тестируемого, уровень ответственности в зависимости от должности и результаты тестирования в виде процентов правильных ответов по каждой из тем, а также окончательный результат как средневзвешенное по всем темам теста. Однако на этом действия компьютера заканчиваются. Далее преподаватель переписывает результаты тестирования в ведомость. Другими словами, разъединены программно-инструментальные средства, имеющие общую цель – обучение и проверка знаний.

Поставим следующую задачу. Создать интерфейс взаимодействия системы Moodle и программы тестирования CES.



Рис.1 - Интерфейс обмена информацией между программами Moodle и CES

Во-первых, допуск к тестированию должны получать только те курсанты, которые успешно сдали все необходимые курсы. В модульной объектно-ориентированной динамической учебной среде Moodle на каждого пользователя - обучаемого (студента, курсанта) создается учетная запись, предоставляющая право пользования данной средой. Обучаемому присваивается шифр (по его зачетной книжке), который является его индивидуальным паролем. По мере прохождения курсов и сдачи соответствующих экзаменов и тестов, результаты заносятся в индивидуальную карточку студента. Таким образом, происходит накопление оценок по всем дисциплинам за все время обучения. Начиная тестирование, например, по программе CES, курсант вводит свой пароль (шифр), который связывает его с базой данных (таблицей с оценками) системы Moodle, и получает разрешение (проще говоря, доступ к прохождению теста), если все его оценки положительны. Естественно, для этой цели требуется создать интерфейс обмена информацией между соответствующими программами.

Протокол передачи данных, т.е. набор соглашений интерфейса логического уровня, которые определяют обмен данными между различными программами, задает единообразный способ передачи сообщений и обработки ошибок при взаимодействии программного обеспечения разнесенной в пространстве аппаратуры, соединенной тем или иным интерфейсом. Таким образом, обеспечивается связь и передача информации Moodle - CES.

После прохождения теста CES результаты тестирования в виде процентов правильных ответов по каждой из тем, а также итоговый процент как средневзвешенное по всем темам теста по каждому тестируемому передается обратной связью в базу данных системы Moodle. Таким образом, результаты тестирования во внешней программе типа CES автоматически используются для общей оценки курсанта. Кроме того, такой способ фиксации результатов тестирования по специальным программам исключает ошибки в учете результатов тестирования, которые, безусловно, возможны при ручном заполнении ведомостей.

Выводы. Модификация модульной объектно-ориентированной динамической учебной среды Moodle с целью ее взаимодействия с программами тестирования моряков решает проблему создания единой базы результатов обучения и тестирования и автоматизирует процесс итоговой аттестации курсантов.

Литература

1. Бобыр Е.И., Радванская Л.Н., Мартинов В.В., Чепурная Ю.В. Комплексная модель адаптивной

- компьютеризированной системы обучения и тестирования на базе семантических сетей // Вестник № 1 (34) ХНТУ, Херсон, 2009г. С. 491-496.
2. Федорук П.И. Адаптивная система дистанционного обучения и контроля знаний на базе интеллектуальных Интернет-технологий /П.И.Федорук. - Киев: Издательско-дизайнерский отдел ЦИТ Прикарпатского национального университета им. Василия Стефанюка, 2008. - 315 с.
 3. Тименко С.В. Образовательные интернет-системы и моделирование знаний /С.В. Тименко//Лаборатория СЕТ.Киев- 2006. http://www.setlab.net/?view=AIED_Overview
 4. Белозубов А.В., Николаев Д.Г. Система дистанционного обучения Moodle. Учебно-методическое пособие. – СПб., 2007. - 108 с.
 5. Агопинов С.В. Средства дистанционного обучения - СПб.: БХВ-Петербург, 2005. -109 с.
 6. Документация Moodle. <http://docs.moodle.org/ru>
 7. Информационный портал Moodle. <http://moodle.org/>

ОНЛАЙН-БАНКІНГ: ПЕРЕВАГИ ТА ПОТЕНЦІЙНІ НЕБЕЗПЕКИ

Кисельова О.Ю.

Україна, м. Київ, НТУУ "КПІ",

Інститут прикладного системного аналізу

New technology age strives to change our ideas about the possibilities of banking services. Virtual banking today saves time and money for both bank and client. By using the Internet client can refill electronic cash-box, repay loans via credit card and pay for mobile, TV, Internet, municipal services. Also new technology has led to the appearance of new banking risks types which must be identified and analyzed.

Світ новітніх інформаційних технологій змінює наші уявлення про можливості банківських послуг. Тепер банки намагаються кожну хвилину бути на зв'язку зі своїм клієнтом. У найближчі роки дистанційні банківські послуги вийдуть на перші позиції. Це пов'язано не тільки з потребою банків знижувати свої витрати, а й продиктоване потребою самих користувачів отримувати доступний, зручний і швидкий канал обслуговування.

На кінець 2013 року [1] інтернет-банкінг в Україні пропонують 38 банків з 50-ти лідерів за активами. Дистанційними можуть стати такі операції як оплата комунальних платежів, поповнення мобільних телефонів, перекази коштів, як між своїми рахунками, так і між картками різних клієнтів. Дистанційні канали можуть замінювати відділення у великих містах, що мають велику частку користувачів інтернету. З іншого боку, в нашій країні є багато людей, чиї фізичні можливості не дозволяють використовувати звичні банківські сервіси. Віртуальний банк - це одне з рішень, яке допоможе людям з обмеженими фізичними можливостями користуватися всім спектром банківських послуг.

Сучасні системи інтернет-банкінгу пропонують наступні опції:

- Оплата комунальних послуг - окрема функція, за допомогою якої платник може перевести гроші за комунальні послуги, вибравши потрібне йому відділення з наданого переліку без самостійного вводу реквізитів.
- Онлайн-депозити - функція інтернет-банкінгу, яка дозволяє відкривати карткові депозити онлайн.
- Управління картою - функція, яка дає можливість заблокувати карту онлайн у випадку її втрати, а також встановлювати, скасовувати і змінювати ліміти на оплату картою в інтернеті.
- Оплата мобільного зв'язку, інтернету, покупка квитків на транспорт, погашення кредитів, придбання полісів автострахування.
- Програми для смартфонів - слід окремо виділити що популярну серед клієнтів функцію. Сьогодні в Україні вже є банки, що створили додатки для мобільного банкінгу на трьох доступних платформах: Android, iOS, Windows.

Віртуальний банк - це реальність вже сьогодні і як приклад можемо навести російський банк «Тінькофф Кредитні Системи» (ТКС) [2]. Бізнес-процеси банку нічим не відрізняються від процесів у будь-якому комерційному банку. Головною відмінною рисою від інших банків є те, що він не має відділень. Банк розвивався дуже швидко з 2010 року і зараз займає 62 - е місце в рейтингу Банку Росії за активами, 47 - е місце по кредитному портфелю, 53- е місце за вкладками фізичних осіб. Проте в схемі, де немає відділень банку, існує ризик, що кредити можуть бути оформлені на крадені документи, адже якісна ідентифікація клієнтів кур'єрами і співробітниками пошти неможлива.

Дійсно, крім всіх позитивних сторін інтернет-банкінгу, залишається питання, як підготуватися до можливих проблем і ризиків при користуванні ним.

Основна небезпека, яка підстерігає користувачів віртуальними банками та інтернет-банкінгом зокрема, це шахрайські платіжні операції. Найбільш слабкою ланкою будь-якої технологічної схеми завжди є людина. Тому шахраї широко використовують різні прийоми, щоб дізнатись логін і пароль клієнтів. До найбільш поширених з них відносяться:

- «Фішинг» - розсилка листів, які провокують одержувачів відправляти свої дані аутентифікації зловмисникам. Популярним є «фішинг» через соціальні мережі, сайти-клони відомих порталів - шахраї, представляючись співробітниками банку, просто запитують дані вашої картки.
- Існують віруси, які вміють знаходити та красти на зараженому комп'ютері дані аутентифікації користувачів для доступу до різних сервісів і здійснювати віддалене управління зараженими комп'ютерами. Тобто користувач може і не помітити, як на його комп'ютері зловмисник здійснює операцію.
- Можливе зламування баз даних інтернет-магазинів, які зазвичай зберігають в незахищеному вигляді інформацію про платежі клієнтів.

Щоб уникнути таких неприємностей, банки продовжують вдосконалювати методи захисту карткових рахунків:

- Електронні методи захисту

Для захисту інтернет-платежів міжнародні платіжні системи використовують технологію 3-DSecure – це додатковий захисний код, який вводиться користувачем при кожній транзакції. Для карт Visa ця технологія має назву «Verified by