

УДК 004.85 (072)

Лещук С. О.

Тернопільський національний педагогічний університет імені Володимира Гнатюка, Тернопіль, Україна

ОКРЕМІ МЕТОДИЧНІ АСПЕКТИ ПІДГОТОВКИ ІТ-ФАХІВЦІВ

DOI: 10.14308/ite000621

Важливим завданням вищих навчальних закладів є підготовка конкурентоздатного професіонала, людини, що легко відчуває себе у ринкових умовах та інформаційному суспільстві. Поряд із володінням інформаційно-комунікаційними технологіями необхідні вміння навчатись у групі, навички роботи над спільними проектами.

Метою статті є продемонструвати можливість організації навчальної діяльності студентів зі змістом і формою роботи, які потребуються сучасною галуззю інформаційних технологій. Автором описано окремі кроки підготовки майбутніх фахівців, яких потребує ІТ-сфера; спеціалістів, які володіють сучасним інструментарієм, розуміють сучасні підходи у програмуванні; вміють працювати у команді над спільним проектом та досягати результату. Розглянуто основні ідеї об'єктно-орієнтованого програмування, методологію управління проектами для гнучкої розробки програмного забезпечення Scrum та можливості використання систем керування версіями, як потужного інструменту, що дає змогу одночасно, без завад один одному, проводити роботу над груповими проектами.

Опис здійснюється на основі проведення спецкурсу для фахівців з інформатики спеціальностей «Прикладна математика», «Інформатика» фізико-математичного факультету (Тернопільський національний педагогічний університет імені Володимира Гнатюка). Зміст матеріалу розширений також ознайомленням з архітектурним шаблоном MVC та узагальненням вмінь роботи з програмним інструментарієм, необхідним для створення веб-проектів. Практичним здобутком навчальної діяльності студентів є розробка соціального веб-проекту з метою підвищення рівня культури суспільства та доброзичливої атмосфери міста.

Ключові слова: інформаційно-комунікаційні технології, проект, об'єктно-орієнтоване програмування, Scrum, системи керування версіями, процес навчання.

Вступ

Підготовка конкурентоздатного професіонала, людини, яка легко відчуває себе у ринкових умовах та інформаційному суспільстві – ось, напевне, важливі завдання вищих навчальних закладів; мета модернізації освіти. Реформування, як правило, відбувається з двох причин: економічної та культурної та супроводжується підвищенням стандартів.

Звичайно, кожна галузь має свою специфіку, та сучасність формує спільні риси кожного алгоритму досягнення успіху. Невід'ємною складовою є формування інформаційних компетентностей та розвиток дивергентного мислення. Дивергентність можна розуміти як обов'язкову умову для творчості (створення оригінальних ідей, що володіють певною цінністю), як здатність знаходити множину можливих способів вирішення задачі та по різному інтерпретувати саму умову задачі.

Зміна парадигми освіти супроводжується розвитком нових методів та форм навчання [1]. Вміння працювати чи навчатись у групі – вже не мета, а необхідність сучасності, адже співпраця – запорука розвитку. Тут варто зазначити, що кожному студентові оптимальний свій власний сценарій навчання у малій чи великій групі, а, можливо, самотійно. Також має місце те, що кожному комфортніше працювати у різний час доби. Отож, було б добре не виставляти «жорстких рамок», а підбирати завдання та форми звіту у такий спосіб, щоб

кожен зміг їх виконати в індивідуальному графіку навчання. Цього можливо досягнути в умовах сучасного етапу інформатизації суспільства.

Таким чином, **метою статті** є продемонструвати можливість організації навчальної діяльності студентів зі змістом і формою роботи, які потребуються сучасною галуззю інформаційних технологій.

Пов'язані роботи

Над проблемами впровадження інформаційно-комунікаційних технологій у навчальний процес працює низка науковців: В. Ю. Биков, Л. І. Білоусова, М. І. Жалдак, В. І. Клочко, Н. В. Морзе, Ю. С. Рамський, Ю. О. Триус, С. А. Раков, З. С. Сайдаметова, О. В. Співаковський та ін. Здобутки цих та багатьох інших учених формують шлях модернізації освіти, спрямовують її на якісно новий рівень. Сюди відносимо розробку освітніх концепцій, методичних систем, засобів і технологій навчання.

У дослідженні [2] О. В. Співаковський та Г. М. Кравцов серед стратегічних завдань впровадження інформаційних технологій в університеті виділяють підготовку майбутніх ІТ-фахівців. Н. В. Морзе досліджує, якою має бути університетська освіта у сучасному суспільстві [3].

Уже понад два десятиліття система освіти України працює у цьому напрямі, зокрема й через підготовку вчителів інформатики. Протягом усього цього часу, однією з основних змістових ліній методичної системи навчання інформатики у педагогічних університетах незаперечно залишається вивчення основ алгоритмізації та програмування. Як зазначають Ю. С. Рамський та М. І. Жалдак, навчання програмування неможливе без фундаментальної підготовки з інформатики майбутніх учителів інформатики, зокрема ґрунтовна підготовка з математики, вміння розв'язувати різні математичні задачі, зокрема з дискретної математики, вміння будувати відповідні математичні й інформаційні моделі та аналізувати їх, знати найбільш відомі задачі дискретної математики та алгоритми їх розв'язування [4].

В. Ю. Биков, М. П. Шишкіна відзначають необхідність застосування особистісно-орієнтованого навчального середовища у процесі підготовки кадрів для індустрії високих технологій [5]. Також, варто зазначити значний педагогічний потенціал технологій хмарних обчислень [6, 7].

Опис спецкурсу

У даній статті описано окремі кроки підготовки майбутніх фахівців, яких потребує ІТ-сфера; спеціалістів, які володіють сучасним інструментарієм, розуміють сучасні підходи у програмуванні; вміють працювати у команді над спільним проектом та досягати результату. Опис здійснюється на основі проведення спецкурсу для студентів п'ятого курсу спеціальностей «Прикладна математика», «Інформатика» фізико-математичного факультету (Тернопільський національний педагогічний університет імені Володимира Гнатюка), на вивчення якого передбачено два кредити (12 лекційних годин та 24 лабораторних).

Кафедра інформатики і методики її викладання оперативно реагує на зміни, що відбуваються в галузі інформаційних технологій. Навчальні плани та програми корегуються з урахуванням сучасних тенденцій розвитку у реальному ІТ-секторі. Основою підготовки фахівців є поєднання ґрунтовної математичної підготовки з професійним володінням сучасними інформаційними технологіями. Взаємодія з можливим працедавцем сприятиме формуванню у наших студентів сучасної моделі кваліфікацій та компетенцій у галузі ІТ. Останні роки відзначаються активізацією науковців кафедри у соціальній сфері, що призвело, зокрема, до збільшення числа партнерів та зміцнення взаємовигідного співробітництва з ними. Росте число шкіл, вищих навчальних закладів, наукових установ, комерційних, громадських, державних та інших організацій з якими кафедра співпрацює на постійній основі. Детальніше про цей та інші напрямки роботи кафедри інформатики можна дізнатися на сайті факультету [8].

Навчальні плани кафедри відповідають державним стандартам та відображають потреби сучасного інформаційного суспільства. Найновіші тенденції відображаються у

вдосконаленні професійно-орієнтованих курсів, розробці нових спецкурсів. Як уже зазначалось, у даній статті описано один з них.

Розглянемо його зміст, методи та форми організації навчальної діяльності.

На початку відбувається знайомство з основними ідеями корпоративного програмування. Ми обрали Scrum – методологію управління проектами для гнучкої розробки програмного забезпечення. Scrum чітко робить акцент на якісному контролі процесу розробки. Ось як про це говориться у самій спільноті: «Scrum підтримує нашу потребу бути людиною на роботі: належати, вчитися, працювати, творити і бути творчим, рости, вдосконалюватися і взаємодіяти з іншими людьми. Іншими словами, Scrum використовує вроджені риси і характеристики людей, щоб дати змогу їм робити великі речі» [9].

Scrum – це основа процесу, яка містить набір методів і попередньо визначених ролей. Обговоривши ролі головних дійових осіб: Scrum Master (той хто опікується процесами, веде їх і працює як керівник проекту), Власник Продукту (людина, що представляє інтереси кінцевих користувачів та інших зацікавлених в продукті сторін) та Команда (розробники), моделюємо роботу у групі студентів, що власне і є Командою. Керівником проекту виступив один із випускників нашого фізико-математичного факультету, а зараз – програміст однієї з ІТ-компаній Тернополя.

Щодо Власника Продукту та змісту самого проекту. Кафедра інформатики виступила ініціатором ідеї розробки соціального проекту «Лист до Святого Миколая». Проект полягав у тому, щоб дізнатись про діток, які можуть не отримати подарунків на довгоочікуване свято через скрутне соціальне положення сім'ї або її відсутність. Відповідні дані збирають в організації Карітас Тернополя (метою Карітасу України є розробка та впровадження національних програм для забезпечення соціальної допомоги та підтримки найбільш потребуючого населення України незалежно від національної приналежності чи віросповідання). Як уже зазначалося, кафедра інформатики співпрацює з громадськими організаціями Тернополя і бере активну участь у житті міста. У межах такої співпраці реалізовано проект «Тролейбус Щастя», як складову проекту «Зроби своє місто кращим». Детальніше про нього можна дізнатись у відкритій групі на сторінках Facebook [10].

Щоб максимально наблизити процес навчання до реальної роботи за методологію Scrum, була організована зустріч із координаторами акції «Миколай про тебе не забуде». Фактично, відбувся етап, що має назву product backlog (документація запитів на виконання робіт), який має найвищу пріоритетність за рівнем вимог до роботи, яка повинна бути виконана. Запити на виконання робіт (backlog items), що визначені протягом наради з планування спринту (sprint planning meeting), переміщуються в етап спринту. Протягом цієї наради Власник Продукту інформує про завдання, які він хоче, аби були виконані. Тоді Команда визначає, скільки зі сформованих запитів вони можуть виконати, щоб завершити необхідні частини протягом наступного спринту. Протягом спринту команда виконує визначений фіксований список завдань (т.з. backlog items). Впродовж цього періоду ніхто не має права змінювати перелік запитів на виконання робіт, що слід розуміти, як заморожування вимог (requirements) протягом спринту.

До експерименту були задіяні групи спеціалістів та магістрів. Усі студенти до п'ятого курсу навчались в одній групі, що сформувало певні зв'язки між ними. Невелика чисельність студентів (13 чоловік) та можливість зручно розмістити пари у розкладі дали змогу легко організувати їх спільну діяльність.

У сучасному суспільстві важливі навички командної роботи, головним критерієм успіху якої є якісне розв'язання поставлених завдань. Команда – це група осіб, у яких є спільна мета і які розуміють, що для досягнення цієї мети необхідна робота кожного з членів групи.

Серед факторів успіху ефективної командної роботи виділимо чітку постановку цілей та завдань, а також правильний підбір складу команди. Через обмежений час (обсяг виділених годин на спецкурс) і бажання витримати методологію Scrum, було вирішено об'єднати «розробників» у три групи. Це дало змогу розділити проект на складові та сформувати чіткий список завдань для кожної групи з подібним розподілом ролей для

кожного учасника. Ми намагались, щоб кожен розробник посів своє оптимальне місце в команді. Ось зміст роботи кожної команди:

- адміністративна частина (перевірка статусу користувачів, опрацювання листів);
- сторінка користувачів (список користувачів, зміна статусу, створення користувача);
- головна сторінка (кнопки входу та реєстрації, форма надсилання листа, перенаправлення).

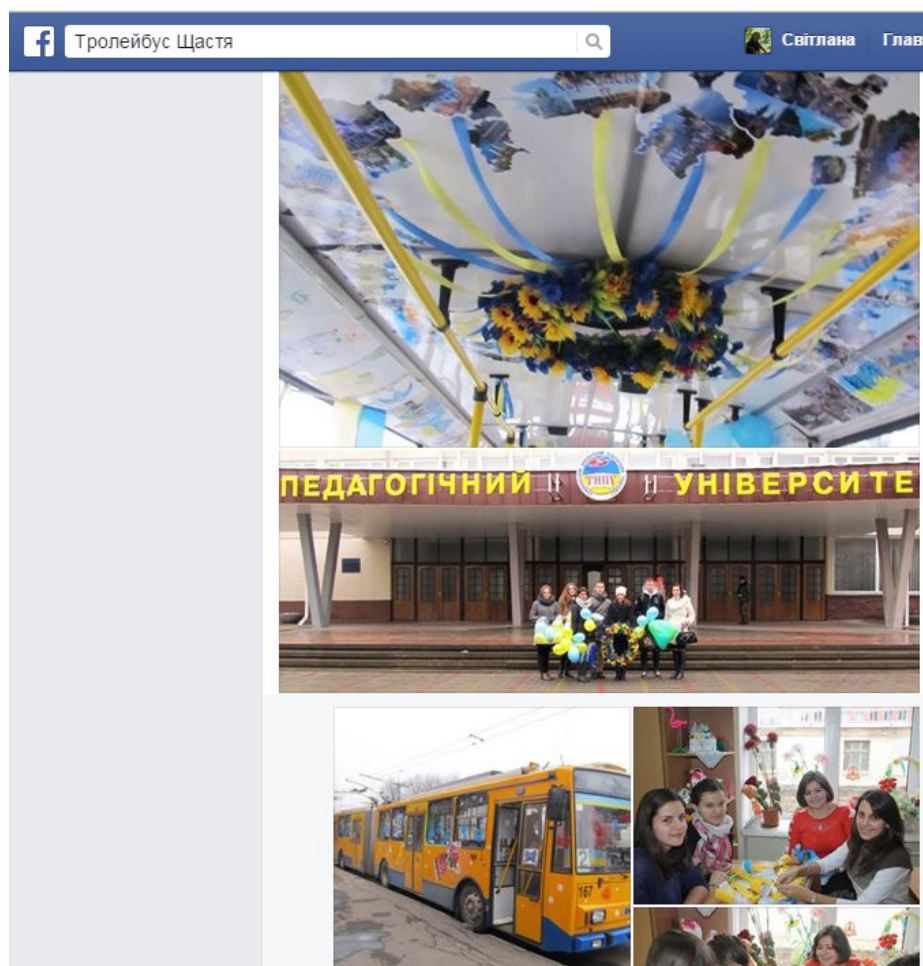


Рис. 1. Проект «Тролейбус Щастя».

Таке об'єднання дало змогу ефективно організувати роботу кожного спринту. На обговореннях, у такому випадку, завжди було декілька рішень, що сприяло розвитку навичок програмування у студентів!

Крім роботи безпосередньо над проектом, завданнями спецкурсу були:

- розуміння принципів об'єктно-орієнтованого програмування;
- ознайомлення з архітектурним шаблоном MVC;
- здобуття навичок використання систем контролю версій;
- узагальнення вмінь роботи з програмним інструментарієм, необхідним для створення веб-проекту.

Звичайно, за роки навчання на фізико-математичному факультеті наші студенти здобувають навички роботи у різноманітних середовищах програмування, зокрема:

- Java-технології;
- Web-програмування;
- бази даних та інформаційні системи;
- мови динамічного програмування;
- моделювання економічних, екологічних та соціальних процесів;

- мультимедійні технології;
- основи паралельного програмування;
- програмування з використанням NET-технологій;
- системне програмування.

На сьогоднішній день в IT-сфері переважають ідеї об'єктно-орієнтованого програмування (ООП). Програми, створені за ООП-підходом, прості та мобільні, їх зручніше модифікувати та підтримувати. Самі ідеї ООП при його грамотному використанні дають змогу програмі бути краще захищеною, ніж це могло бути передбачено під час її створення.

Технологія ООП володіє трьома основними перевагами:

- вона проста для розуміння: ООП дає змогу думати категоріями повсякденних об'єктів;
- надійність і простота для супроводу – правильне проектування забезпечує простоту розширення і модифікації об'єктно-орієнтованих програм. Модульна структура дає змогу вносити незалежні зміни у різні частини програми, зводячи до мінімуму ризик помилок програмування;
- прискорений цикл розробки – завдяки модульності різні компоненти об'єктно-орієнтованих програм можна легко використовувати в інших програмах, що зменшує надмірність коду і знижує ризик помилок при копіюванні.

Важливим завданням курсу є розуміння специфіки ООП, яка заснована на:

- інкапсуляції (механізм, який об'єднує дані й обробляє їх код як єдине ціле). Завдяки включенню дрібних елементів у крупніший об'єкт, з яким працює програміст, відбувається спрощення роботи завдяки виключенню другорядних деталей. Це дає змогу зосередитись на розв'язання конкретних завдань. В ООП інкапсуляція забезпечується класами, об'єктами і різними засобами вираження ієрархічних зв'язків між ними;
- поліморфізмі (дає змогу використовувати одні й ті ж імена для схожих, але технічно різних завдань). Головним у поліморфізмі є можливість маніпулювати об'єктами шляхом створення стандартних інтерфейсів для схожих дій;
- успадкування (механізм, при якому один об'єкт може набувати властивостей іншого об'єкта). Точна копія об'єкта доповнюється унікальними властивостями, які характерні тільки для похідного об'єкта.

Варто ще сказати про абстракцію, яка, напевне, була в програмуванні завжди, починаючи з записів першого в історії програміста Ади Лавлейс. З тих пір люди безперервно створювали у своїх програмах абстракції, часто маючи для цього лише найпростіші засоби. З точки зору програмування, абстракція – це правильне розділення програми на об'єкти. Зазвичай, будь-яку велику програму можна представити у вигляді взаємодіючих об'єктів десятками способів. Абстракція дає змогу відібрати головні характеристики і опустити другорядні.

Для закріплення розуміння принципів ООП було проведено низку дидактичних вправ для формування на інтуїтивному рівні уявлень про основні сутності даного підходу. Зокрема, наводили приклади з життя:

- об'єктів (математичне представлення сутності реального світу або предметної області, яке використовується для моделювання);
- класів (загальна сутність, яка може бути визначена як сукупність елементів), до яких вони відносяться;
- властивостей чи атрибутів (пропозиційна функція, визначена на довільному типі даних), якими вони володіють;
- відповідних методів чи функцій (операція, яка визначена над об'єктами того чи іншого класу).

Різноманітні комбінації, обговорення, дискусії дали змогу чітко сформулювати основні поняття та конструкції. Практикувався й аналіз відібраних у мережі ресурсів, наприклад [11].

Наступним етапом було вивчення архітектурного шаблону MVC. Модель – вигляд – контролер (англ. Model-View-Controller, MVC) – архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення. Цей шаблон поділяє систему на три частини: модель даних, вигляд даних та керування. Застосовується для відокремлення даних (модель) від інтерфейсу користувача (вигляду) так, щоб зміни інтерфейсу користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача.

Мета шаблону – гнучкий дизайн програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонентів програми. Крім того, використання цього шаблону у великих системах сприяє певній впорядкованості їх структури і робить їх зрозумілишими завдяки зменшенню складності.

Зрозуміти особливості MVC вдалось безпосередньо в процесі роботи над проектом. На рисунках 2-4 продемонстровано застосування обраного шаблону для розробки коду.

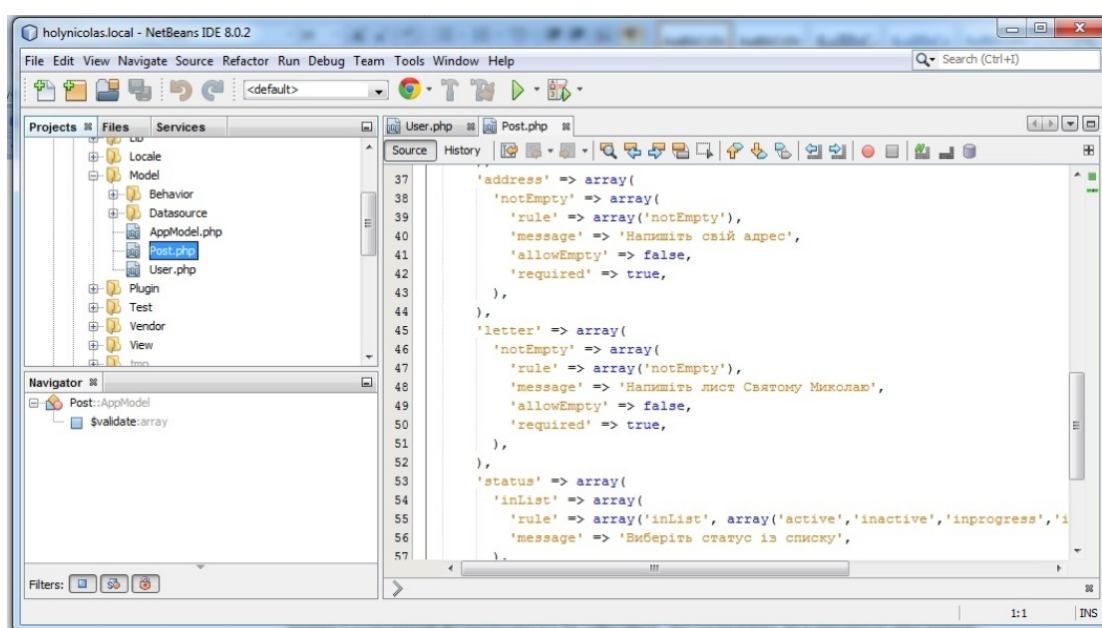


Рис. 2. Model Post.

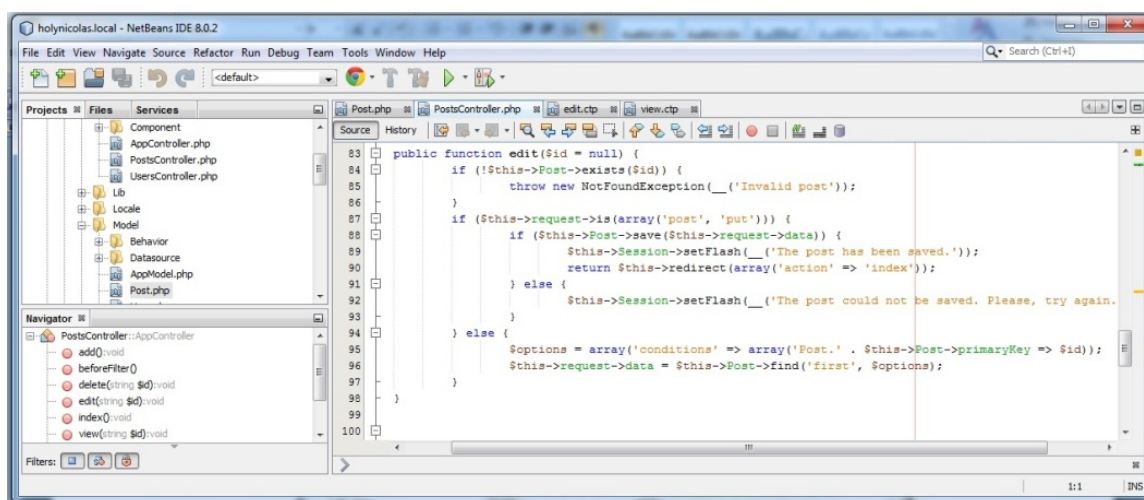


Рис. 3. Controller Posts Controller.

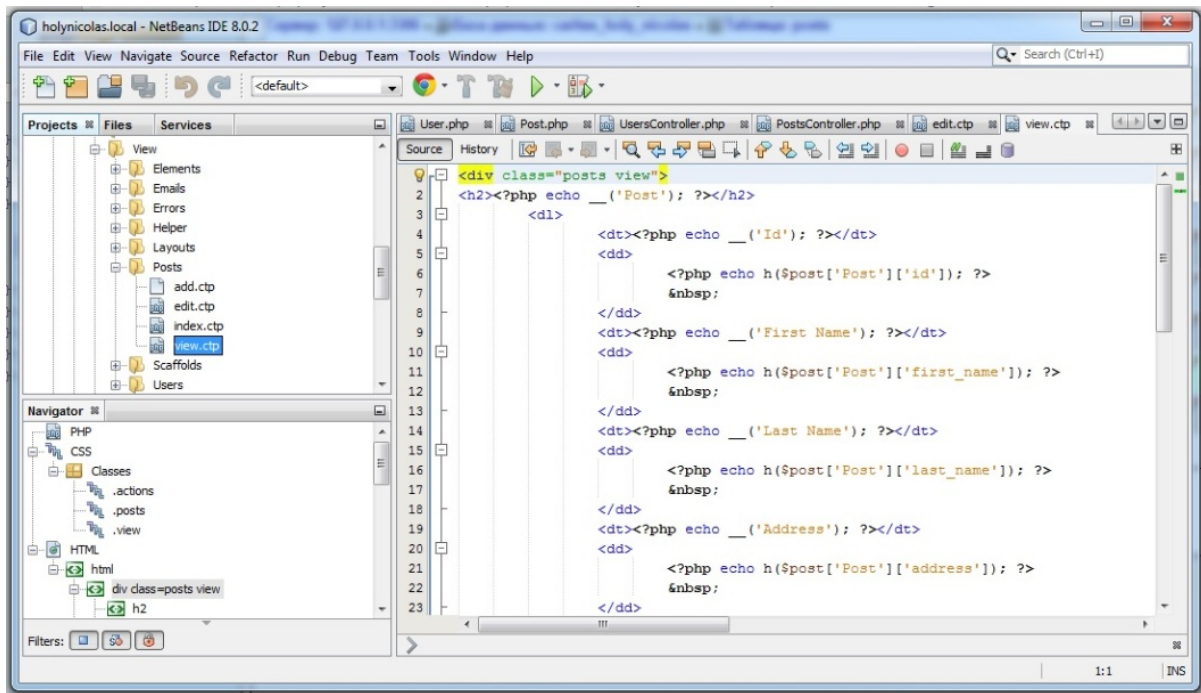


Рис. 4. View Posts.

Узагальнимо функції компонентів шаблону:

- Model – зберігання даних і забезпечення інтерфейсу до них (інкапсулює ядро даних і основний функціонал з їх обробки, не залежить від процесу введення або виведення даних; «логіка» програмного продукту);
- View – представлення цих даних користувачеві в довільному форматі (компонент виводу може мати декілька взаємопов'язаних областей, наприклад, різні таблиці і поля форм, в яких відображаються дані);
- Controller – контроль над компонентами, обробка даних (отримує сигнали у вигляді реакції на дії користувача (зміна положення курсору миші, натиснення кнопки або введення даних в текстове поле) і повідомляє про зміни компоненту Модель).

Така внутрішня структура в цілому поділяє систему на самостійні частини і розподіляє відповідальність між різними компонентами. Варто зазначити, що неможливо уникнути в MVC коду, що дублюється. Це могло б бути проблемою для широкомасштабного проекту, а ознайомлення з HMVC для нас може бути перспективою.

Моделі в основному використовуються для керування відповідною таблицею бази даних і правил взаємодії з нею. У більшості випадків, кожна таблиця в базі даних буде відповідати одній моделі. У нашому проекті, як вже було показано вище, описано дві моделі, таким чином, база даних містить дві таблиці *users* та *posts* (див. Рис. 5).

Оскільки результат проекту – веб-проект, то інструментами програмування обрано скриптову мову програмування для генерації HTML-сторінок на стороні веб-сервера PHP. Ми використовували *phpMyAdmin* – веб-застосунок з відкритим кодом на мові PHP із графічним веб-інтерфейсом для адміністрування систем керування базами даних MySQL. Застосування *phpMyAdmin* дає змогу через браузер здійснювати адміністрування сервера MySQL, запускати запити SQL, переглядати та редагувати вміст таблиць баз даних.

Програма розповсюджується під ліцензією GNU General Public License і тому деякі інші розробники інтегрують його у свої розробки, наприклад XAMPP (багатофункціональна збірка веб-сервера, що містить Apache, MySQL, інтерпретатор скриптів PHP, мову програмування Perl та велику кількість додаткових бібліотек, що дають змогу запуснути повноцінний веб-сервер) [12] чи OpenServer.

Робота над проектом вимагала базових знань та навиків роботи з PHP, MySQL тощо. При необхідності, учасники Команди могли опрацювати окремі матеріали, зокрема [13, 14].

The screenshot shows the phpMyAdmin interface. The left sidebar displays a database tree with 'caritas_holy_nicolas' selected, containing tables 'posts' and 'users'. The main area shows the structure of the 'posts' table, with a detailed view of the 'users' table structure overlaid below it.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
1	id	int(11)			Нет	Нет	AUTO_INCREMENT	
2	first_name	varchar(20)	utf8_general_ci		Нет	Нет		
3	last_name	varchar(20)	utf8_general_ci		Нет	Нет		
4	address	varchar(255)	utf8_general_ci		Нет	Нет		
5	letter	text	utf8_general_ci		Нет	Нет		
6	status	enum('active', 'inactive', 'inprogress', 'invalida')	utf8_general_ci		Нет	Нет		
7	created	datetime			Нет	Нет		
8	modified	datetime			Нет	Нет		

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
1	id	int(11)			Нет	Нет	AUTO_INCREMENT	
2	email	varchar(40)	utf8_general_ci		Нет	Нет		
3	password	varchar(40)	utf8_general_ci		Нет	Нет		
4	role	enum('santa', 'admin')	utf8_general_ci		Нет	Нет		
5	created	datetime			Нет	Нет		
6	modified	datetime			Нет	Нет		

Рис. 5. Структура таблиць бази даних проекту.

Багато ситуацій моделювалося під час вивчення спецкурсу, та обмежений аудиторний час був основною відмінністю від організації роботи програмістів у реальних умовах. Доречним вирішенням цього питання стало використання систем керування версій [15;16]. Це дало змогу планувати періоди спринту, не прив'язуючись до розкладу занять, а лише розраховувати на відповідальне ставлення кожного члена команди до роботи. Кожного разу при виконанні частини коду учасником необхідно було оновити робочу копію, модифікувати проект та зафіксувати зміни.

Системи керування версіями зазвичай використовуються при розробці програмного забезпечення для відстеження, документування та контролю над поступовими змінами в електронних документах, над якими одночасно працюють декілька програмістів. Кожна версія позначається унікальною цифрою чи літерою, зміни документу занотуються. Зазвичай, також зберігається автор зробленої зміни та її час. Інструменти для контролю версій входять до складу багатьох інтегрованих середовищ розробки. Отож, зазначені засоби є потужним інструментом, що дає змогу одночасно, без завад один одному, проводити роботу над груповими проектами.

Використання репозитаріїв для зберігання коду та історії змін до нього є необхідним навиком у роботі програміста. Основна ознака централізованих репозитаріїв коду (Version Control System, VCS) полягає в тому, що може бути лише один центральний репозитарій коду для проекту, куди усі розробники поміщають свій код (приклад, SVN). Звичайно, у такому випадку повинен бути безперебійний доступ до сервера, на якому розміщений репозитарій.

Розподілені репозитарії коду (Distributed Version Control System, DVCS) дають змогу, крім центрального репозитарія, також мати кожному розробнику власний локальний репозитарій, і взагалі скільки завгодно копій / клонів репозитарію та автономно працювати з кожним із них. Для регулярних комітів (процедур поміщення коду) не потрібен постійний доступ до Інтернету, можна робити записи у локальний репозитарій, а вже коли з'явиться Інтернет – синхронізувати усі необхідні зміни на віддалені репозитарії. Найбільш

поширеними розподіленими репозитаріями коду вважаються Git (підтримка github.com) та Mercurial (підтримка bitbucket.com).

На заняттях ми спробували організувати роботу із використанням ресурсів Bitbucket (див. Рис. 6). Це досить складна система, що дає можливість розробникам ефективно управляти вихідним кодом. Лише безпосередня робота з системою контролю версій сприятиме оволодінню цією технологією. Корисними є інтерактивні курси для вивчення Git [17, 18].

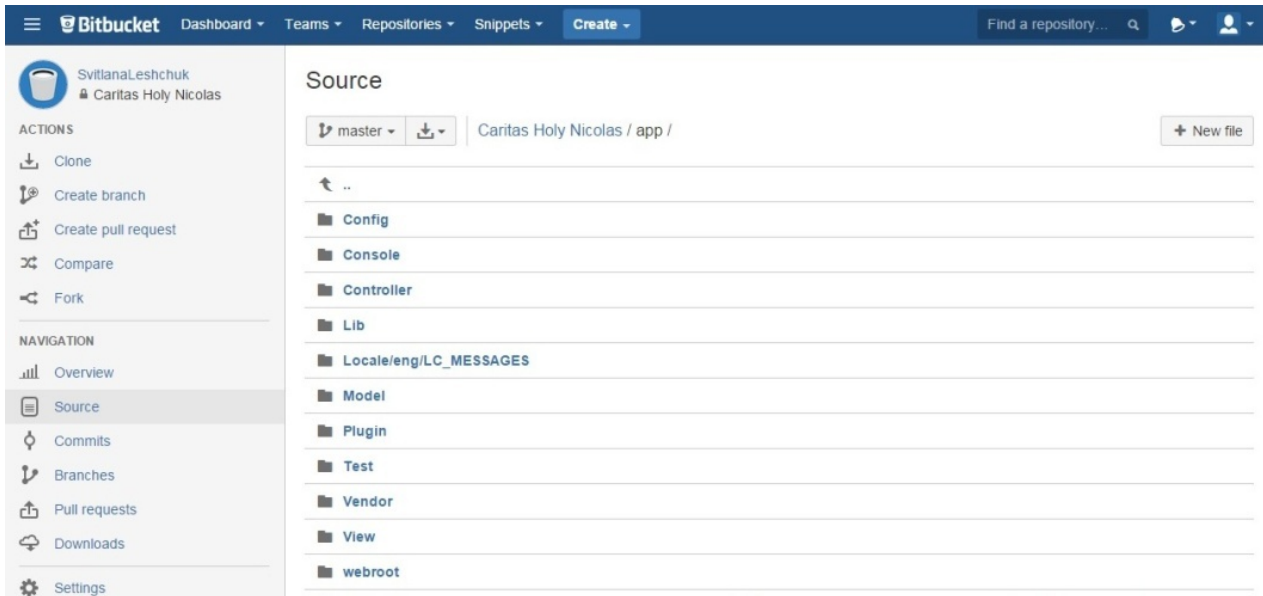


Рис. 6. Репозитарій на Bitbucket.

Крім того, використовували ресурси CakePHP [19], зокрема для відлагодження проекту (див. Рис. 7).

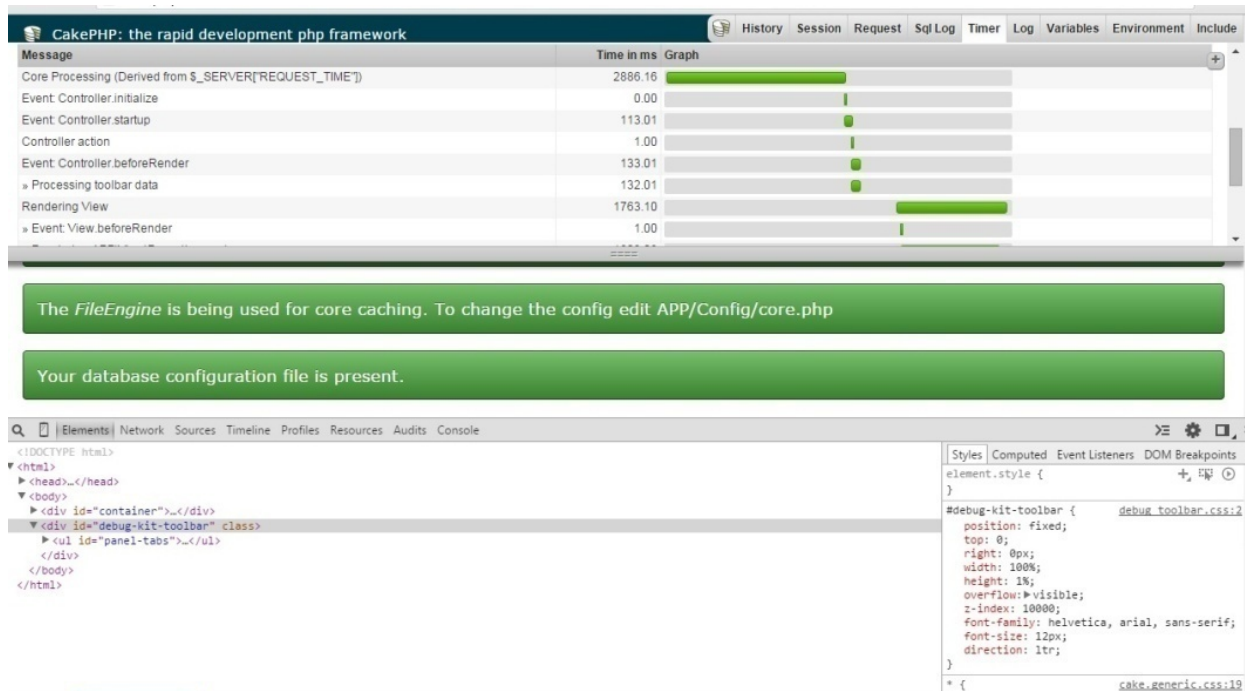


Рис. 7. Відлагодження проекту.

CakePHP – це безкоштовний, з відкритими вихідними кодами, фреймворк для швидкої розробки додатків на PHP. Деякі переваги його використання:

- активне і дружелюбне співтовариство;
- ліцензія;
- повна підтримка як php4, так і php5;
- повна взаємодія з базою даних;
- авто-генерування коду;
- архітектура MVC;
- вбудована верифікація даних;
- система швидких шаблонів;
- компоненти, що автоматизують роботу з поштовими повідомленнями, Cookie, Безпекою, Сесіями і Запитами;
- автоматична обробка даних;
- підтримка багатомовності готового web-додатку;
- робота з будь-якого каталогу сайту, майже без необхідності змінювати конфігурацію Apache.

Провівши спецкурс, удалось отримати проміжний результат проектної роботи і здобути низку необхідних для програмістської діяльності навиків. Розроблений веб-продукт дає змогу збирати листи від дітей Тернопілля, з метою формування бази даних тих, хто потребує «участі Помічників Миколая» (початкові сторінки показані на Рис. 8). Подібні проекти сприяють підвищенню рівня культури суспільства та доброзичливої атмосфери міста. Робота над проектом триває.

Рис. 8. Форма написання листа та входу адміністратора.

Педагогічний експеримент

З метою уточнення змісту спецкурсу та системи оцінювання навчальних досягнень студентів за умов кредитно-модульної системи на пошуковому етапі експерименту було використано метод експертних оцінок.

Дослідження полягає у тому, що основні питання змісту курсу ми пронумерували за зростанням ознаки. Експертами були обрані провідні програмісти та керівники ІТ-відділів Тернополя. Їм ми запропонували вказати місце змістових тем, необхідних для реалізації проекту (значення «1» присвоювалось найбільш важливій на думку експерта темі, «10» – найменш важливий).

Для унеможливлення психологічної підказки, яка могла б вплинути на вибір експертом певної послідовності ранжування, пропонувані теми у картці опитування (табл. № 1) були розміщені у випадковій послідовності.

Таблиця № 1.

Картка опитування з переліком тем

№ теми	Тема	Місце
1.	Основи об'єктно-орієнтованого програмування мовою PHP. Архітектурний шаблон MVC.	
2.	Спільна розробка веб-проектів з використанням систем контролю версій.	
3.	Розробка дизайну веб-сторінки.	
4.	Робота з бібліотеками JavaScript (на прикладі JQUERY).	
5.	Робота з СУБД MySQL у мові PHP.	
6.	Пошукова оптимізація веб-проекту.	
7.	Основи розробки проектів у галузі програмування.	
8.	Програмування мовою PHP з використанням фреймворків (на прикладі CakePHP).	
9.	Технології верстання веб-сторінок.	
10.	Засоби для тестування веб-сторінок.	

Результати опитування експертів подано у Таблиці № 2.

Таблиця № 2.

Результати опитування

Експерт	Тема №1	Тема №2	Тема №3	Тема №4	Тема №5	Тема №6	Тема №7	Тема №8	Тема №9	Тема №10
1	5	4	8	2	3	7	9	1	6	10
2	4	3	6	1	5	8	10	7	2	9
3	2	4	10	3	1	8	9	6	7	5
4	3	5	7	1	4	9	8	2	10	6
5	1	5	7	2	3	10	8	6	4	9
6	4	7	5	6	2	8	9	1	3	10
7	1	5	8	3	6	10	9	7	2	4
8	5	6	8	7	1	9	10	3	2	4
9	2	3	9	4	1	6	10	5	7	8
10	4	6	7	3	2	8	9	1	5	10
11	3	7	6	2	4	8	10	5	9	1
12	3	6	8	4	5	7	10	2	1	9
13	1	5	6	2	7	8	10	3	4	9
14	6	3	7	1	5	9	8	2	4	10
15	5	4	10	3	8	7	9	2	1	6
16	1	2	7	5	3	6	10	8	4	9
17	2	1	8	4	5	9	10	3	6	7
18	4	8	6	1	7	10	9	3	2	5
19	2	7	10	3	4	6	9	5	1	8
20	1	3	8	4	7	10	9	2	5	6

Найбільш очевидним показником оцінювання теми є її сумарний ранг, який визначений усіма експертами ($S_j = \sum_{i=1}^{20} R_{i,j}$, де R – показник, виставлений i-тим експертом j-ій теми).

Проте такі сумарні ранги будуть об'єктивними, якщо між експертами є певний рівень погодження. Ступінь такого погодження описує коефіцієнт конкордації W [20], який визначається так:

Для кожної теми знаходимо різницю сумарних показників із їх середнім значенням:

$$d_j = \sum_{i=1}^m R_{i,j} - 0,5 \cdot m \cdot (n+1) \quad (1)$$

де m – число експертів, n – кількість тем, $j = \overline{1, n}$.

Знаходимо суму квадратів величин, отриманих із співвідношення (1)

$$S(d^2) = \sum_{j=1}^n d_j^2 = \sum_{j=1}^n \left[\sum_{i=1}^m R_{i,j} - 0,5 \cdot m \cdot (n+1) \right]^2 \quad (2)$$

Максимальне значення величини $S(d^2)$ ($S_{\max}(d^2) = \frac{1}{12} \cdot m^2 (n^3 - n)$) досягається у випадку, якщо всі експерти ранжуватимуть критерії (теми) однаково.

Коефіцієнт конкордації дорівнює:

$$W = \frac{S(d^2)}{S_{\max}(d^2)} = \frac{12 \cdot S(d^2)}{m^2 (n^3 - n)} \quad (3)$$

Із Таблиці № 2 знаходимо значення величин сумарних рангів та d_j (табл. № 3):

Таблиця № 3.

<i>Результати ранжування</i>										
	Тема №1	Тема №2	Тема №3	Тема №4	Тема №5	Тема №6	Тема №7	Тема №8	Тема №9	Тема №10
S_j	59	94	151	61	83	163	185	74	85	145
d_j	-51	-16	41	-49	-27	53	75	-36	-25	35

Із співвідношень (2), (3) обчислюємо коефіцієнт конкордації $W=0,583$. Ця величина завжди знаходиться між нулем та одиницею. Якщо $W=0$, то зв'язку між ранжуваннями експертів не існує, якщо $W=1$, то ранжування збігаються повністю. Отриманий нами коефіцієнт $W=0,583$ суттєво відрізняється від нуля, тому можна стверджувати, що між експертами існує об'єктивне погодження.

Проте таке значення коефіцієнта W не є критерієм об'єктивності, оскільки могло бути отримане внаслідок випадкового виставлення рангів тієї чи іншої теми.

Величина $m \cdot (n-1) \cdot W$ розподілена за законом χ^2 із $n-1$ ступенем вільності.

Використовуючи співвідношення $\chi_w^2 = \frac{12 \cdot S(d^2)}{m \cdot n \cdot (n+1)}$, знаходимо значення $\chi_w^2 = 104,99$.

Порівнюючи його із табличним для $\nu = n-1 = 9$ ступенів вільності і для рівня значущості $\alpha = 0,05$, отримуємо $\chi_w^2 = 104,99 > \chi_t^2 = 16,92$, з чого робимо висновок, що між висновками експертів існує узгодженість.

Отже, з результатів опитування видно, що найбільш важливими для проведення спецкурсу є теми:

1. Основи об'єктно-орієнтованого програмування мовою PHP. Архітектурний шаблон MVC.
2. Робота з бібліотеками JavaScript (на прикладі JQUERY).
3. Програмування мовою PHP з використанням фреймворків.

Проводячи експеримент, ми розуміли, що ранжовані експертами теми належать до компетенцій різних фахівців у галузі розробки веб-сайтів. Проте виконуючи такий проект, студенти матимуть змогу опанувати базові уміння, а також визначитися із майбутньою спеціалізацією.

Висновки та перспективи

Отже, організувавши таким чином навчальний процес, ми моделюємо роботу багатьох напрямків IT-галузі. Це дає змогу освоїти сучасний інструментарій, зрозуміти сучасні

підходи у програмуванні, навчитись працювати у команді над спільним проектом. На такому спецкурсі студенти знайомляться зі змістом і формами роботи, які є актуальними в сучасній сфері інформаційних технологій.

Експериментальне дослідження дало змогу визначити основні тематичні напрямки, освоєння яких доцільне у процесі підготовки ІТ-фахівців (основи об'єктно-орієнтованого програмування мовою PHP; архітектурний шаблон MVC; робота з бібліотеками JavaScript; програмування мовою PHP з використанням фреймворків).

Даний проект сприятиме вдосконаленню навчального процесу, а також розробці нових підходів для підготовки конкурентоздатного професіонала, фахівця, який легко відчуває себе у ринкових умовах та інформаційному суспільстві.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Sir Ken Robinson Changing Paradigms [Електронний ресурс]. – Режим доступу: <https://www.thersa.org/discover/videos/event-videos/2008/06/changing-paradigms/>
2. Співаковський О. В. Цілі, задачі та забезпечення стратегічного плану впровадження інформаційних технологій в концепції розвитку університету / О. В. Співаковський, Г. М. Кравцов // Інформаційні технології в освіті. – 2012. – № 13. – С. 09-22.
3. Морзе Н. В. Яким має бути «розумний» університет у «розумному» суспільстві? / Н. В. Морзе // Сучасні стратегії університетської освіти : якісний вимір [Текст]: матер. міжнарод. наук.-практ. конференції 28-29 березня 2012 / Київський університет імені Бориса Грінченка, Міністерство освіти і науки, молоді та спорту України ; ред. В. О. Огнев'юк; редкол.: В. П. Андрущенко, Л. Л. Хоружа. – Київ : Київ ун-т ім. Б. Грінченка, 2012. – С. 87-98.
4. Жалдак М. І., Рамський Ю. С. Становлення і розвиток методичної системи навчання інформатики в школах і педагогічних університетах України / М. І. Жалдак, Ю. С. Рамський // Науковий часопис НПУ імені М. П. Драгоманова. Серія №2. Комп'ютерно-орієнтовані системи навчання: Зб. наук. праць /Редрада. – К.: НПУ імені М. П. Драгоманова, 2012. – № 13 (20). – С. 24-41.
5. Shyshkina. M. Innovative models of education and training of skilled personnel for high tech industries in Ukraine / .M. Shyshkina, V. Bykov // Informational Technologies in Education. – 2013. – № 15. – P. 19-29.
6. Tryus Y., Kachala T. Cloud technologies in management and educational process of Ukrainian technical universities // Informational Technologies in Education. - Kherson: KSU. - Vypusk19. – 2014. – P. 22-33.
7. Олексюк В. Досвід організації віртуальних лабораторій на основі технологій хмарних обчислень / В. Олексюк // Інформаційні технології в освіті. – 2014. – № 20. – С. 128-138.
8. Сайт фізико-математичного факультету Тернопільського національного педагогічного університету імені Володимира Гнатюка [Електронний ресурс]. – Режим доступу: <http://fizmat.tnpu.edu.ua/>
9. What is Scrum? [Електронний ресурс]. – Режим доступу: <https://www.scrum.org/Resources/What-is-Scrum>
10. Тролейбус Щастя. [Електронний ресурс]. – Режим доступу: <https://www.facebook.com/groups/1479799778951651/?fref=ts>
11. Основи ООП для початківців. Відеокурс. [Електронний ресурс]. – Режим доступу: <http://yellow-duck.net/catalog/5/>
12. XAMPP Installersand Downloadsfor Apache Friends. [Електронний ресурс]. – Режим доступу: <https://www.apachefriends.org/ru/index.html>
13. Справочное руководство по MySQL. [Електронний ресурс]. – Режим доступу: http://www.mysql.ru/docs/man/Creating_database.html
14. Tutorial – W3Schools. [Електронний ресурс]. – Режим доступу: <http://www.w3schools.com/default.asp>
15. Michael Cochez, Ville Isomottonen, Ville Tirronen, Jonne Itkonen The Use of Distributed Version Control Systems in Advanced Programming Courses [Електронний ресурс]. – Режим доступу: <http://ceur-ws.org/Vol-1000/ICTERI-2013-p-221-235.pdf>
16. Струк О. О. Використання систем керування версіями у командній роботі над проектами під час вивчення програмування / О. О. Струк, С. П. Струк // Матеріали регіонального науково-

- практичного семінару. «Шляхи удосконалення навчального процесу в контексті інноваційних змін в системі вищої освіти». Тернопіль. – 2011 р. – С. 23-26.
17. Git How To. [Електронний ресурс]. – Режим доступу: <http://githowto.com/ru>
 18. Ежедневная работа с Git. [Електронний ресурс]. – <http://habrahabr.ru/post/174467/>
 19. CakePHP: the rapid development php framework. [Електронний ресурс]. – Режим доступу: <http://cakephp.org/>
 20. Математико-статистические методы экспертных оценок. 2-е изд. пер. и доп./ Бешелев С.Д., Гурвич Ф.Г.– М.:Статистика, 1980. – 263 с.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. 1. Ken Robinson Changing Paradigms. (b.d.). Retrieved from <https://www.thersa.org/:https://www.thersa.org/discover/videos/event-videos/2008/06/changing-paradigms/>
2. Spivakovskij, O. V., & Kravcov, G. M. (2012). Cili, zadachi ta zabezpechennja strategichnogo planu vprovadzennja informacijnih tehnologij v koncepcii rozvitku universitetu. Informacijni tehnologii v osviti, str. 09-22.
3. Morze, N. V. (2012). Jakim maє buti «rozumnij» universitet u «rozumnomu» suspil'stvi? (str. 87-98). Kiiiv: Kiiiv un-t im. B. Grinchenka.
4. Zhaldak, M. I., & Rams'kij, Ju. S. (2012). Stanovlennja i rozvitok metodichnoї sistemi navchannja informatiki v shkolah i pedagogichnih universitetah Ukraїni. Naukovij chasopis NPU imeni M. P. Dragomanova. Serija №2. Komp'juterno-orientovani sistemi navchannja: Zb. nauk. prac', str. 24-41.
5. Shyshkina, M., & Bykov, V. (2013). 5. Innovative models of education and training of skilled personnel for high tech industries in Ukraine. Informational Technologies in Education, str. 19-29.
6. Tryus, Y., & Kachala, T. (2014). Cloud technologies in management and educational process of Ukrainian technical universities. Informational Technologies in Education, str. 22-33.
7. Oleksjuk, V. (2014). Dosvid organizacii virtual'nih laboratorij na osnovi tehnologij hmarnih obchislen'. Informacijni tehnologii v osviti, str. 128-138.
8. Sajt fiziko-matematichnogo fakul'tetu Ternopil'skogo nacional'nogo pedagogichnogo universitetu imeni Volodimira Gnatjuka . (b.d.). Retrieved from fizmat.tnpu.edu.ua/: <http://fizmat.tnpu.edu.ua/>
9. What is Scrum? (b.d.). Retrieved from www.scrum.org/: <https://www.scrum.org/Resources/What-is-Scrum>
10. Trolejbus Shhastja. (b.d.). Retrieved from www.facebook.com/:https://www.facebook.com/groups/1479799778951651/?fref=ts
11. Osnovi OOP dlja pochatkivciv. Videokurs. (b.d.). Retrieved from [yellow-duck.net/: http://yellow-duck.net/catalog/5/](http://yellow-duck.net/catalog/5/)
12. XAMPP Installers and Downloads for Apache Friends. (b.d.). Retrieved from [www.apachefriends.org/: https://www.apachefriends.org/ru/index.html](http://www.apachefriends.org/:https://www.apachefriends.org/ru/index.html)
13. Spravochnoe rukovodstvo po MySQL. (b.d.). Retrieved from [www.mysql.ru/: http://www.mysql.ru/docs/man/Creating_database.html](http://www.mysql.ru/:http://www.mysql.ru/docs/man/Creating_database.html)
14. Tutorial – W3Schools. (b.d.). Retrieved from [www.w3schools.com/: http://www.w3schools.com/default.asp](http://www.w3schools.com/:http://www.w3schools.com/default.asp)
15. Michael Cochez, Ville Isomottonen, Ville Tirronen, Jonne Itkonen The Use of Distributed Version Control Systems in Advanced Programming Courses. (b.d.). Retrieved from [ceur-ws.org/: http://ceur-ws.org/Vol-1000/ICTERI-2013-p-221-235.pdf](http://ceur-ws.org/:http://ceur-ws.org/Vol-1000/ICTERI-2013-p-221-235.pdf)
16. Struk, O. O., & Struk, S. P. (2011). Viktoristannja sistem keruvannja versijami u komandnij roboti nad proektami pid chas vivchennja programuvannja. Materiali regional'nogo naukovopraktichnogo seminaru. «Shljahi udoskonalennja navchal'nogo procesu v konteksti innovacijnih zmin v sistemi vishhoї osviti», str. 23-26.
17. Git How To. (b.d.). Retrieved from githowto.com/: <http://githowto.com/ru>
18. Ezhdnevnaja rabota s Git. (b.d.). Retrieved from [habrahabr.ru/: http://habrahabr.ru/post/174467/](http://habrahabr.ru/:http://habrahabr.ru/post/174467/)
19. CakePHP: the rapid development php framework. (b.d.). Retrieved from [cakephp.org/: http://cakephp.org/](http://cakephp.org/:http://cakephp.org/)
20. Beshelev, S. D., & Gurvich, F. G. (1980). Matematiko-statisticheskie metody jekspertnyh ocenok. 2-e izd. per. i dop. Moskva: Statistika.

Стаття надійшла до редакції 21.02.17

Svitlana Leshchuk

Ternopil V. Hnatyuk National Pedagogical University, Ternopil, Ukraine

SOME METHODOLOGICAL ASPECTS OF TRAINING IT PROFESSIONALS

The important task of higher educational establishments is preparation of competitive professional, man that easily feels in market conditions and informative society. Next to possessing of informatively-communication technologies necessary abilities to study in a group, skills of prosecution of general projects.

The aim of the article is to show possibility of organization of educational activity of students with maintenance and form works that is required by modern industry of information technologies. An author is describe separate steps preparations of future specialists, that is required by a IT-sphere; specialists that own a modern tool are understood by modern approaches in programming; able to work in a command above a general project and to arrive at a result. The basic ideas of the object-oriented programming are considered, methodology of management projects for the flexible Scrum and possibilities of the use of control system by versions software development, as a powerful instrument that gives an opportunity simultaneously, swimmingly to each other, to conduct the prosecution of group projects.

Description comes true on the basis of realization of the special course for specialists on the informatics of specialities «Applied mathematics», «Informatics» of physics-mathematics faculty (Ternopil Volodymyr Hnatyuk National Pedagogical University). The table of contents of material is extended also by an acquaintance with the architectural template of MVC and generalization of abilities of work with a programmatic tool, by a necessity for creation of web-projects. The practical achievement of educational activity of students is development of social web-project with the aim of increase of level of culture of society and benevolent atmosphere of city.

Keywords: ICT, project design, object-oriented programming, Scrum, version control systems, Teaching Process.

Лещук С. А.

Тернопольский национальный педагогический университет имени Владимира Гнатюка, Тернополь, Украина

НЕКОТОРЫЕ МЕТОДИЧЕСКИЕ АСПЕКТЫ ПОДГОТОВКИ ИТ-СПЕЦИАЛИСТОВ

Важной задачей высших учебных заведений является подготовка конкурентоспособного специалиста, человека, который легко чувствует себя в рыночных условиях и информационном обществе. Наряду с владением информационно-коммуникационными технологиями необходимые умения учиться в группе, навыки работы над совместными проектами.

Целью статьи является продемонстрировать возможность организации учебной деятельности студентов с содержанием и формой работы, которые требуются современной отраслью информационных технологий. Автором описано отдельные шаги подготовки будущих специалистов, в которых нуждается ИТ-сфера; специалистов, которые обладают современным инструментарием, понимают современные подходы в программировании; умеют работать в команде над общим проектом и достигать результата. Рассмотрены основные идеи объектно-ориентированного программирования, методологию управления проектами для гибкой разработки программного обеспечения Scrum и возможности использования систем управления версиями, как мощного инструмента, что позволяет одновременно, без помех друг другу, проводить работу над групповыми проектами.

Описание осуществляется на основе проведения спецкурса для студентов специальностей «Прикладная математика», «Информатика» физико-математического факультета (Тернопольский национальный педагогический университет имени Владимира Гнатюка). Содержание материала расширено также ознакомлением с архитектурным шаблоном MVC и обобщением умений работы с программным инструментарием, необходимым для создания веб-проектов. Практическим достижением спецкурса является

разработка социального веб-проекта с целью повышения уровня культуры общества и доброжелательной атмосферы города.

Ключевые слова: информационно-коммуникационные технологии, проект, объектно-ориентированное программирование, Scrum, системы управления версиями, процесс обучения.