

200 РОКІВ
ОСВІТНІХ ТРАДИЦІЙ



Том 2



**ТЕЗИ
70-ої наукової конференції
професорів, викладачів, наукових
працівників, аспірантів та студентів університету**

**ПОЛТАВСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ
УНІВЕРСИТЕТ ІМЕНІ ЮРІЯ КОНДРАТЮКА**

Міністерство освіти і науки України
Північно-Східний науковий центр НАН України та МОН України
Полтавський національний технічний університет
імені Юрія Кондратюка

Тези

70-ої наукової конференції професорів,
викладачів, наукових працівників, аспірантів
та студентів університету

Том 2

23 квітня – 18 травня 2018 р.

Полтава 2018

УДК 043.2
ББК 448ЛО

*Розповсюдження та тиражування без офіційного дозволу
Полтавського національного технічного університету
імені Юрія Кондратюка заборонено*

Редакційна колегія:

Онищенко В.О.	д.е.н., проф., ректор Полтавського національного технічного університету імені Юрія Кондратюка
Сівіцька С.П.	к.е.н., доц., проректор з наукової та міжнародної роботи
Гришко В.В.	д.е.н., проф., директор навчально-наукового інституту фінансів, економіки та менеджменту
Іваницька І.О.	к.х.н., доц., декан гуманітарного факультету
Нестеренко М.П.	д.т.н., проф., декан будівельного факультету
Матвієнко А.М.	к.т.н., доц., заступник директора навчально-наукового інституту нафти і газу
Муравльов В.В.	к.т.н., доц., в.о. декана архітектурного факультету
Шульга О.В.	д.т.н., доц., директор навчально-наукового інституту інформаційних технологій та механотроніки

Тези 70-ої ювілейної наукової конференції професорів, викладачів, наукових працівників, аспірантів та студентів університету. Том 2. (Полтава, 23 квітня – 18 травня 2018 р.) – Полтава: ПолтНТУ, 2018. – 380 с.

У збірнику тез висвітлені результати наукових досліджень професорів, викладачів, наукових працівників, аспірантів та студентів університету.

помилку стимулює то, що модель підганяється під кінцеву вибірку. Цю помилку можна розділити на дві складові, зсув моделі та дисперсію. Зсув є середня помилка, яку дана процедура навчання моделі робить на різних наборах даних. Дисперсія відповідає чутливості процедури моделювання до конкретного вибору даних [1].

Розпізнавання помилок аналізу є надзвичайно важливим етапом який може значно збільшити ефективність аналізу та його застосування, адже буде мати більш коректні дані на виході які допоможуть полегшити прогнозування та підвищити його точність. Ці дії призведуть до підвищення якості прийнятих рішень на основі прогнозування часових рядів.

Література

1. С.П. Альошин, О.О. Бородіна Навчальний посібник «Інтелектуальний аналіз даних» для студентів денної та заочної форми навчання: 6.050101 – «Комп'ютерні науки» – Полтава: ПолтНТУ, 2014. 187 с

2. Аналіз часових рядів [Електронний ресурс].– Режим доступу: https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B0%D0%BB%D1%96%D0%B7_%D1%87%D0%B0%D1%81%D0%BE%D0%B2%D0%B8%D1%85_%D1%80%D1%8F%D0%B4%D1%96%D0%B2

UNIT ТЕСТУВАННЯ ЯК СПОСІБ ВИЯВИТИ СЛАБКІ МІСЦЯ В ANDROID ПРОГРАМІ

Модульне тестування, або юніт-тестування (англ. Unit testing) – процес в програмуванні, що дозволяє перевірити на коректність окремі модулі вихідного коду програми.

Ідея полягає в тому, щоб писати тести для кожної нетривіальної функції або методу. Це дозволяє досить швидко перевірити, чи не призвело чергову зміну коду до регресії, тобто до появи помилок в місцях програми, які були протестовані, а також полегшує виявлення й усунення таких помилок [3].

Модульне тестування має не тільки плюси, а й мінуси. До переваг модульного тестування відносять: по-перше, модульне тестування дозволяє бути впевненим, що через виправлення одного бага не з'явився інший. Оскільки тести працюють в автоматичному режимі, завжди можете відстежити описану вище ситуацію (провівши, тим самим, регресійні тестування). Ще один плюс, який, за великим рахунком, групується з першим, – unit-тести дозволяють безболісно проводити рефакторинг коду (тобто зміна самого коду без якого б то не було зміни його функціональності). Багато хто навіть стверджує, що модульні тести заохочують рефакторинг коду розробниками [1].

Другий плюс полягає в тому, що unit-тести служать деяким чином документацією до коду й путівником по ньому. І справді, тести можуть розповісти, як та чи інша функції повинні реагувати на різні вхідні параметри – опис цієї поведінки часто по не цілком зрозумілих причин відсутня в коментарях, що залишаються програмістами.

Третій плюс полягає в тому, що застосування unit-тестів дає більш якісне відділення інтерфейсу від реалізації. Оскільки в рамках одного тесту перевіряється один клас, то все "порочні" зв'язку з іншими класами, не передбачені архітектурою додатки, впливають на поверхню й безжальним чином розриваються.

До переваг модульного тестування відносять: по-перше, якщо тестуєте кожен елемент програмного коду окремо від інших, ніхто не зможе сказати напевно, що станеться, коли всю цю купу-малу змішайте воедино. Тобто, не змішайте, а інтегруєте в єдине ціле. Тому не обійтися без додаткових інтеграційних тестів. Але це ще півбіди. Наступний мінус полягає в тому, що при недостатньо продуманій архітектурі проекту тести

можуть заважати зміни коду, а не сприяти йому. Ситуація не така вже й рідкісна, а тому й далеко не остання в списку неприємностей.

Але найгірше те, що навіть тести, успішно виконані й перевиконані після внесення усіляких змін, не дають ніякої гарантії, що в коді немає помилок. І справа тут навіть не в підгонці тестів під код, а в людському факторі. Іноді дещо змінивши значення вхідних параметрів в рамках тестів програміст може, сам того не бажаючи, пропустити помилку в тестованому коді або, гірше того, внести її туди. Але, від помилок, пов'язаних з людським фактором, не застрахований ніхто і ніколи.

Вище я згадував, що є спеціальні інструменти, які допомагають програмісту писати unit-тести, нижче наведено основні із них.

Назви більшості популярних фреймворків для unit-тестування утворені від назви мови, на якій пишеться програмний код, і слова "unit". Наприклад, для Java це JUnit, для .NET'овських мов - NUnit, для Delphi - DUnit, для C ++ - CPPUnit, для PHP - PHPUnit, для Python - PyUnit, для ActionScript - AsUnit ... Ну і так далі. А якщо не знаходиться фреймворк для unit-тестування для тієї мови, на якій пишете, швидше за все, ця мова або ще не придумали, або вже ніде не використовують, або це Асемблер.

Що ж, як бачите, в самій ідеї unit-тестування немає нічого складного. Та й в практиці, мабуть, теж. Потрібно тільки знайти час та сили на вивчення тестіровочного фреймворка для використовуваного вами мови програмування і для того, щоб писати самі тести.

Література

1. KV.by [Електронний ресурс].– Режим доступу: <https://www.kv.by/archive/index2009421107.htm>.
2. Про Тестинг - Тестирование Программного Обеспечения [Електронний ресурс].– Режим доступу: <http://www.protesting.ru/testing/levels/component.html>.
3. Хабрахабр [Електронний ресурс].– Режим доступу: <https://habrahabr.ru/post/169381/>.

<i>О.І. Тиртишніков, Т.І. Хавер</i> ОЦІНЮВАННЯ ПРОДУКТИВНОСТІ ВІДЕОПІДСИСТЕМИ ПЕРСОНАЛЬНОГО КОМП'ЮТЕРА	138
<i>Ю.Л.Поночовний, В.С.Воронянський</i> АНАЛІЗ ЗАСТОСУВАННЯ МЕТОДОЛОГІЙ ТА CASE ЗАСОБІВ ДЛЯ УПРАВЛІННЯ ПРОЦЕСАМИ ФУНКЦІОНУВАННЯ НАВЧАЛЬНОГО ЗАКЛАДУ	139
<i>К.О. Колісніченко, Ю.Л. Поночовний</i> АНАЛІЗ ВИТРАТ ЕЛЕКТРОЕНЕРГІЇ ДЛЯ РОЗВ'ЯЗАННЯ ПАРАЛЕЛЬНИХ ЗАДАЧ НА ПЛАТФОРМІ ЛОКАЛЬНОЇ МАШИНИ ТА ВІРТУАЛЬНОГО СЕРВЕРУ	140
<i>Ю.Л.Поночовний, Р.О. Киричок</i> АНАЛІЗ МЕТОДІВ ПРОЕКТУВАННЯ ЦИФРОВИХ СИСТЕМ ВІДЕОСПОСТЕРЕЖЕННЯ.....	141
<i>В.А. Горошко, Ю.Л. Поночовний</i> ОРГАНІЗАЦІЯ ЛОКАЛЬНОЇ МЕРЕЖІ ПІДПРИЄМСТВА ТОВ «КАІР-СЕРВІС»	142
<i>В.Г. Канівець, М.О. Нестеренко, С.В. Сомов</i> МОДЕЛЬ РЕЄСТРАЦІЇ ПОДІЙ У КОМПЛЕКСНІЙ СИСТЕМІ ЗАХИСТУ ІНФОРМАЦІЇ	143
<i>Пузь Д.О., С.В. Сомов</i> МОДЕЛЮВАННЯ ІГРОВОГО ОБ'ЄКТА З ВИКОРИСТАННЯМ ПРОГРАМИ 3D MAX	144
<i>Л.М. Дегтярьова, О.В. Хорошун</i> ОСОБЛИВОСТІ ВИКОРИСТАННЯ ІНФОРМАЦІЙНОГО ПОРТАЛУ ДЛЯ СИСТЕМАТИЗАЦІЇ ТА ЗБЕРІГАННЯ ІНФОРМАЦІЇ	146
<i>Л.М. Дегтярьова, В.А.Супрун</i> ВИКОРИСТАННЯ МІКРОКОНТРОЛЕРІВ ДЛЯ ВДОСКОНАЛЕННЯ ФУНКЦІЙ ПРИБОРІВ ТА СИСТЕМ	148
<i>Л.М. Дегтярьова, В.В. Касян</i> ВПРОВАДЖЕННЯ ЕЛЕКТРОННОЇ КОРПОРАТИВНОЇ ПОШТОВОЇ РОЗСИЛКИ НА ПІДПРИЄМСТВІ.....	150
<i>Л. М. Дегтярьова, В. А. Хелемес</i> МОДЕЛІ ТА МЕТОДИ АНАЛІЗУ ПОВЕДІНКИ ВІДВІДУВАЧІВ ВЕБ-САЙТІВ	152
<i>Л.М. Дегтярьова, І.Ю. Яблуновський</i> АСПЕКТИ ТА ВИМОГИ ПРОЕКТУВАННЯ ХМАРНОЇ ІНФРАСТРУКТУРИ ЗБЕРІГАННЯ ДАНИХ	154
<i>А. С. Янко, В. В. Авдєєв</i> ПЕРЕВАГИ МЕРЕЖ ДОСТУПУ З ВИКОРИСТАННЯМ РАДІОРЕЛЕЙНИХ ЛІНІЙ ЗВ'ЯЗКУ.....	156
<i>А. С. Янко, В. С. Горбань</i> ОЦІНКА ЗАГРОЗ ЦІЛІСНОСТІ ІНФОРМАЦІЇ КОРПОРАТИВНОЇ МЕРЕЖІ.....	158

І.В. Ромашко ВИКОРИСТАННЯ ПРОТОКОЛУ VTR ДЛЯ МАСШТАБУВАННЯ ЛОКАЛЬНИХ МЕРЕЖ	160
І.В. Ромашко ЗАСТОСУВАННЯ ТЕХНОЛОГІЇ АГРЕГУВАННЯ КАНАЛІВ ДЛЯ ПІДВИЩЕННЯ ПРОПУСКНОЇ ЗДАТНОСТІ МЕРЕЖІ.....	160
І.Я. Гудзенко, Сокол Г. В. ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАВАДОСТІЙКОГО КОДУВАННЯ НА ОСНОВІ C++	161
В.А. Василевська, Часовських І.С., Г.В. Сокол, Т.В. Буряк WI-FI РАДІО З ІНТЕРАКТИВНИМ БЕЗДРОТОВИМ УПРАВЛІННЯМ.....	162
А.В. Виноградова, Г.В. Сокол, Т.В. Буряк АНАЛІЗ ТЕХНІЧНИХ РІШЕНЬ ДЛЯ РЕАЛІЗАЦІЇ РОБОТИЗОВАНИХ КОМПЛЕКСІВ	164
В.Ю. Литвиненко, Г.В. Сокол ПОРІВНЯЛЬНИЙ АНАЛІЗ ЛІЦЕНЗІЙНИХ ТА БЕЗКОШТОВНИХ КОДЕКІВ ДЛЯ ОБРОБКИ БАГАТОВИМІРНИХ СИГНАЛІВ	166
О.В. Мосієнко, Г.В. Сокол АНАЛІЗ АУДІОПЛЕСРІВ ОБРОБКИ ОДНОВИМІРНИХ СИГНАЛІВ.....	168
В.Р. Ткаченко, Г.В. Сокол АНАЛІЗ ПРОБЛЕМ МОНІТОРИНГУ СИСТЕМ SMART HOUSE	169
СЕКЦІЯ КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В СИСТЕМ	
Ляхов О.Л., Демиденко М.І., Фурсова Н.А. АРХІТЕКТУРА РОЗПОДІЛЕНОЇ АВТОМАТИЗОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ КРУВАННЯ НАВЧАЛЬНИМ ПРОЦЕСОМ ВНЗ	170
С.П. Альошин, О.О. Бородіна НЕЙРОМЕРЕЖЕВИЙ ПРЕДИКТИВНИЙ МЕТОД ОПТИМІЗАЦІЇ В ЗАДАЧІ БАГАТОФАКТОРНОГО АНАЛІЗУ	173
С.О. Зайка, А.Т. Лобурець РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ РОЗРАХУНКУ ДІАГРАМ СТАНУ ТЕРМОДИНАМІЧНИХ СИСТЕМ	176
О.А. Руденко, М.І. Демиденко, А.А. Швидкий ПРОГРАМНИЙ МОДУЛЬ «ОБЛІК УСПІШНОСТІ» АВТОМАТИЗОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ НАВЧАЛЬНИМ ПРОЦЕСОМ	178
Гайтан О.М., Горошко А.І. АНАЛІЗ СИСТЕМ ПЕРЕВІРКИ НАУКОВИХ ТА АКАДЕМІЧНИХ ТЕКСТІВ НА ОРИГІНАЛЬНІСТЬ.....	180