

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЖИТОМИРСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Т.А. ВАКАЛЮК

Технології тестування програм

*навчально-методичний посібник
для студентів напрямку 6.040302 Інформатика**

Житомир 2013

УДК 004.415.53:004.052

ББК 32.973.26-018.2

В14

*Затверджено Вченою радою Житомирського державного університету
імені Івана Франка протокол № 3 від 25 жовтня 2013 р*

Рецензенти:

Сейдаметова З.С. – доктор педагогічних наук, професор

Медведєв М.Г. – кандидат фізико-математичних наук, доцент

Вакалюк Т.А.

В14 Технології тестування програм. Навчально-методичний посібник для студентів напряму 6.040302 Інформатика*. – Житомир: Вид-во ЖДУ, 2013. – 96 с.

Посібник призначений для використання студентами під керівництвом викладача на лекціях, практичних та лабораторних заняттях. Посібник містить лекційний курс та лабораторний практикум з технологій тестування програм. Викладений матеріал відповідає діючій програмі з предмету "Технології тестування програм" для спеціальності 6.040302 "Інформатика"

Для студентів напряму 6.040302 "Інформатика*" вищих педагогічних закладів, вчителів інформатики загальноосвітніх шкіл.

УДК 004.415.53:004.052

ББК 32.973.26-018.2

Зміст

Вступ

ТЕОРЕТИЧНІ ВІДОМОСТІ ІЗ ТЕХНОЛОГІЙ ТЕСТУВАННЯ ПРОГРАМ

Тема 1. Основні етапи розв'язування задач з програмування.

Тема 2. Тестування і налагодження програм.

Тема 3. Налagodження і тестування програмних засобів.

Тема 4. Тестування програми як "чорного" та "білого" ящика.

Тема 5. Принципи тестування

Тема 6. Типи помилок і ручні методи тестування

Тема 7. Стратегії білого ящика.

Тема 8. Стратегії чорного ящика

Тема 9. Низхідне та висхідне тестування

Тема 10. Проектування і виконання тесту

ЛАБОРАТОРНИЙ ПРАКТИКУМ

Загальні вимоги до виконання лабораторних робіт

Лабораторна робота №1

Лабораторна робота №2

Лабораторна робота №3

Лабораторна робота №4

Лабораторна робота №5

Лабораторна робота №6.

Лабораторна робота №7.

Лабораторна робота №8.

Лабораторна робота №9.

Лабораторна робота №10.

Лабораторна робота №11

Тестові завдання

Тема 1. Основні етапи розв'язування задач з програмування

Тема 2. Тестування і налагодження програм

Тема 3. Налagodження і тестування програмних засобів

Тема 4. Тестування програми як "чорного" та "білого" ящика

Тема 5. Принципи тестування

Тема 6. Типи помилок і ручні методи тестування.

Тема 7. Стратегії білого ящика

Тема 8. Стратегії чорного ящика

Тема 9. Низхідне та висхідне тестування

Тема 10. Проектування і виконання тесту

Рекомендована література

Вступ

*Тестування виявляє лише наявність, але ніяк не відсутність помилок.
Е.Дейкстра*

Програма, вільна від помилок – це абстрактне теоретичне поняття.

Мало в якому вигляді діяльності існує стільки можливостей для помилок, як у програмуванні. Одним з критеріїв професійної майстерності програмістів є їх спроможність виявляти та виправляти власні помилки.

Відомо, що при створенні типового програмного проекту близько 50% загального часу і більше 50% загальної вартості витрачається на тестування розробленої програми або системи. Будь-який програміст може похвалити "добре" написаним кодом, модулем, класом, але, як правило, він практично нічого не може сказати, наскільки повно протестувано цей код. Багато готові лаяти інших розробників, вказуючи на їхні помилки, збої їх програмного забезпечення, забуваючи про свої.

За останні роки технології створення програмного забезпечення (ПЗ) стали основою різних розділів комп'ютерних наук як засіб подолання складності, що притаманна сучасним програмним системам. Але жодна з відомих технологій не здатна в корені змінити сумного факту – помилки в програмі зустрічаються завжди. Ми знаходимо їх за допомогою тестування, а усуваємо їх за допомогою налагодження. Програмісти-початківці не вміють цього робити, досвідчені програмісти – уміють, але помилки роблять усі без виключень. Як не важко з цим змиритись, але хороші програмісти знають, що основний час при програмуванні буде витрачений на тестування та налагодження.

Саме тому дисципліна "Технології тестування програм" входить до базових дисциплін циклу професійної та практичної підготовки як невід'ємна складова освіти студентів за спеціальністю 6.040302 "Інформатика".

Мета даного видання є допомогти студентам оволодіти теоретичними знаннями та практичними навиками роботи з управлінням якістю програмного забезпечення на етапах життєвого циклу, тестування з метою створення корисних і працездатних програмних продуктів.

У запропонованому посібнику подано теоретичний матеріал та практичні приклади з базових тем курсу. В матеріалах розглядаються критерії та методи тестування програмного забезпечення на різних етапах життєвого циклу. Подано перелік лабораторних робіт, які необхідно виконати протягом вивчення дисципліни.

ТЕОРЕТИЧНІ ВІДОМОСТІ ІЗ ТЕХНОЛОГІЙ ТЕСТУВАННЯ ПРОГРАМ

Тема 1. Основні етапи розв'язування задач з програмування.

Мета: ознайомити студентів з основними етапами розв'язування задач з програмування.

Професійна спрямованість: дана лекція є складовою частиною професійної підготовки вчителя інформатики до викладання в школі.

Основні поняття: задача, задача з програмування, основні етапи розв'язання задач.

План лекції:

1. Основні етапи розв'язання задач з програмування.
 - 1.1. Постановка задачі.
 - 1.2. Опис алгоритму.
 - 1.3. Складання програми.
 - 1.4. Тестування та налагодження програми.
 - 1.5. Експлуатація програми.

Обрані методи: теоретичні, словесні, наочні, пояснювально-ілюстративний.

Наочність: схематичні зображення, схеми, що використовуються в ході викладання матеріалу.

Питання по темі для самостійного вивчення:

1. Що таке експлуатація програми?
2. Назвіть випадки коли був супровід програмістами розробленого ПЗ для нового налагодження програми?

Запитання для самоаналізу та самоперевірки:

1. Які ви знаєте етапи розв'язання задачі з програмування?
2. Що таке налагодження?
3. Що таке тестування?

Тема 2. Тестування і налагодження програм.

Мета: ознайомити студентів з основними поняттями тестування та налагодження.

Професійна спрямованість: дана лекція є складовою частиною професійної підготовки вчителя інформатики до викладання в школі.

Основні поняття: тестування, налагодження, логічна помилка, семантична помилка, синтаксична помилка.

План лекції:

1. Налагодження.
2. Деякі типові помилки в програмах.

3. Види помилок
4. Захисне програмування.
5. Методологічні принципи налагодження програм.
6. Тестування.
7. Деякі принципи тестування програм.
8. Деякі прийоми тестування.
9. Види тестових даних.
10. Приклади складання тестів

Обрані методи: теоретичні, пояснювально-ілюстративний.

Наочність: схеми, що використовуються в ході викладання матеріалу.

Питання по темі для самостійного вивчення:

1. Захисне програмування.
2. Методологічні принципи налагодження програм.

Запитання для самоаналізу та самоперевірки:

1. Що таке налагодження?
2. Що таке тестування?

Тема 3. Налагодження і тестування програмних засобів.

Мета: вивчити основні поняття курсу.

Професійна спрямованість: дана лекція є складовою частиною професійної підготовки вчителя інформатики до викладання в школі.

Основні поняття: налагодження, тестування, надійність.

План лекції:

1. Налагодження ПЗ.
2. Тестування ПЗ.
3. Надійність ПЗ.

Обрані методи: теоретичні, пояснювально-ілюстративний.

Наочність: схеми, що використовуються в ході викладання матеріалу.

Питання по темі для самостійного вивчення:

1. Надійність ПЗ.

Запитання для самоаналізу та самоперевірки:

1. Дайте означення налагодження ПЗ.
2. Дайте означення тестування ПЗ.
3. Які ви знаєте основні заповіді по тестуванню програм?
4. Як ви розумієте надійність ПЗ?

Тема 4. Тестування програми як "чорного" та "білого" ящика.

Мета: ознайомити студентів з поняттям технології тестування чорного та білого ящика.

Професійна спрямованість: дана лекція є складовою частиною професійної підготовки вчителя інформатики до викладання в школі.

Основні поняття: тестування, чорний ящик, білий ящик.

План лекції:

1. Тестування програми як чорного ящика.
2. Тестування програми як білого ящика.

Обрані методи: теоретичні, пояснювально-ілюстративний.

Наочність: схематичні зображення.

Запитання для самоаналізу та самоперевірки:

1. Тестування програми як чорного ящика.
2. Тестування програми як білого ящика.

Тема 5. Принципи тестування

Мета: ознайомити студентів з принципами тестування.

Професійна спрямованість: дана лекція є складовою частиною професійної підготовки вчителя інформатики до викладання в школі.

Основні поняття: функціональне тестування.

План лекції:

1. Принципи тестування

Обрані методи: лекція-бесіда.

Наочність: схематичні зображення.

Питання по темі для самостійного вивчення:

1. Характеристики хорошого тесту.

Запитання для самоаналізу та самоперевірки:

1. Сформулюйте характеристики хорошого тесту.
2. Придумайте, яким чином два програмісти, що створюють одну програму, можуть протестувати її, не порушуючи принципів тестування.
3. Покажіть і доведіть, що абсолютне тестування неможливе на конкретній програмі.
4. Дайте визначення поняттю «тестування».
5. Перерахуйте принципи тестування.
6. Поясніть, чому тестування є деструктивним процесом.
7. Чому при тестуванні необхідно бажати, щоб програма дала збій?
8. Чому в ділянці коду, де було виявлено найбільше помилок, може міститися ще більша їх кількість?
9. До чого веде планування тесту в припущенні відсутності помилок?
10. Що дає перевірка програми на неправильних вхідних даних?
11. Навіщо необхідно зберігати тести, якщо програма вже випущена?

Тема 6. Типи помилок і ручні методи тестування

Мета: ознайомити студентів з типами помилок.

Професійна спрямованість: дана лекція є складовою частиною професійної підготовки вчителя інформатики до викладання в школі.

Основні поняття: помилка.

План лекції:

1. Класифікація помилок.
2. Первинне виявлення помилок.
3. Інспекції і наскрізні перегляди.
 - 3.1. Інспекції вихідного тексту.
 - 3.2. Наскрізні перегляди.
 - 3.3. Перевірка за столом.
4. Список питань для виявлення помилок при інспекції.
 - 4.1. Помилки звернення до даних.
 - 4.2. Помилки опису даних.
 - 4.3. Помилки обчислень.
 - 4.4. Помилки при порівняннях.
 - 4.5. Помилки в передачах управління.
 - 4.6. Помилки інтерфейсу.
 - 4.7. Помилки введення-виведення.

Обрані методи: теоретичні, пояснювально-ілюстративний.

Наочність: схеми, що використовуються в ході викладання матеріалу.

Питання по темі для самостійного вивчення:

1. Класифікація помилок по часу їх появи.
2. Класифікація помилок за ступенем порушення логіки.

Запитання для самоаналізу та самоперевірки:

1. Дайте визначення терміна «помилка».
2. Наведіть класифікацію помилок по часу їх появи.
3. Наведіть класифікацію помилок за ступенем порушення логіки.
4. Які помилки (в різних класифікаціях) бувають в програмах на мові C і коли вони з'являються?
5. Які мови виявляють помилки структурного набору?
6. Визначте вид помилки: `if ((x > 3) && (x < 2)) ...`
7. Які типові помилки зустрічаються в програмах?
8. У чому полягає сутність інспекції?
9. Які етапи включає метод наскрізного перегляду програми?
10. Наведіть приклад помилки звернення до даних.
11. Наведіть приклад помилки опису даних.
12. Наведіть приклад помилки інтерфейсу.
13. Наведіть приклад помилки передачі управління.
14. Наведіть приклад помилки при порівняннях.
15. Наведіть приклад помилки обчислення.
16. Наведіть приклад помилки введення-виведення.

Тема 7. Стратегії білого ящика.

Мета: ознайомити студентів з стратегіями тестування білого ящика.

Професійна спрямованість: дана лекція є складовою частиною професійної підготовки вчителя інформатики до викладання в школі.

Основні поняття: тестування, покриття операторів, покриття рішень, покриття умов, покриття рішень/умов.

План лекції:

1. Покриття операторів.
2. Покриття рішень.
3. Покриття умов.
4. Покриття рішень/умов.
5. Комбінаторне покриття умов

Обрані методи: словесні, наочні, пояснювально-ілюстративний.

Наочність: схеми, що використовуються в ході викладання матеріалу.

Питання по темі для самостійного вивчення:

1. Комбінаторне покриття умов.

Запитання для самоаналізу та самоперевірки:

1. Опишіть процес тестування методом покриття операторів.
2. Опишіть процес тестування методом покриття умов.
3. Опишіть процес тестування методом покриття рішень.
4. Опишіть процес тестування методом покриття рішень / умов.
5. Опишіть процес тестування методом комбінаторного покриття умов.

Тема 8. Стратегії чорного ящика

Мета: ознайомити студентів з технологією тестування чорного ящика.

Професійна спрямованість: дана лекція є складовою частиною професійної підготовки вчителя інформатики до викладання в школі.

Основні поняття: еквівалентне розбиття, аналіз граничних значень, застосування функціональних діаграм.

План лекції:

1. Еквівалентне розбиття.
 - 1.1. Виділення класів еквівалентності.
 - 1.2. Побудова тестів.
2. Аналіз граничних значень.
3. Застосування функціональних діаграм.
 - 3.1. Зауваження.
4. Припущення про помилку.

Обрані методи: теоретичні, пояснювально-ілюстративний.

Наочність: схеми, що використовуються в ході викладання матеріалу.

Питання по темі для самостійного вивчення:

1. Застосування функціональних діаграм.

Запитання для самоаналізу та самоперевірки:

1. Опишіть процес тестування методом аналізу граничних значень.
2. Опишіть процес тестування методом еквівалентного роздроблення.
3. Опишіть процес тестування методом функціональних діаграм.
4. Опишіть процес тестування методом припущення про помилку.

Тема 9. Низхідне та висхідне тестування

Мета: ознайомити студентів з низхідним та висхідним тестуванням.

Професійна спрямованість: дана лекція є складовою частиною професійної підготовки вчителя інформатики до викладання в школі.

Основні поняття: низхідне тестування, висхідне тестування.

План лекції:

1. Стратегії.
2. Низхідне та висхідне тестування.
 - 2.1. Низхідне тестування.
 - 2.2. Висхідне тестування.

Обрані методи: теоретичні, пояснювально-ілюстративний.

Наочність: схематичні зображення.

Питання по темі для самостійного вивчення:

1. Стратегії.

Запитання для самоаналізу та самоперевірки:

1. У чому полягає метод висхідного тестування?
2. У чому полягає метод спадного тестування?
3. Порівняйте методи висхідного і низхідного тестування.

Тема 10. Проектування і виконання тесту

Мета: ознайомити студентів з проектуванням і виконанням тесту.

Професійна спрямованість: дана лекція є складовою частиною професійної підготовки вчителя інформатики до викладання в школі.

Основні поняття: тесту, проектування.

План лекції:

1. Проектування і виконання тесту

Обрані методи: теоретичні, пояснювально-ілюстративний.

Наочність: схематичні зображення.

Запитання для самоаналізу та самоперевірки:

1. У чому суть проектування тесту?

ЛАБОРАТОРНИЙ ПРАКТИКУМ

Загальні вимоги до виконання лабораторних робіт

Порядок виконання лабораторних робіт

1. Ознайомитись з темою, метою роботи та засвоїти матеріал теоретичної частини.
2. Вибрати відповідний варіант (за вказівкою вчителя) та ознайомитись із завданням.
3. Розробити програму для виконання поставленого завдання.
4. Розробити систему тестів, яка б повністю та достовірно перевіряла правильність роботи програми (нормальні, нульові, критичні та екстремальні умови).
5. Зберегти результати виконаної роботи в індивідуальній робочій папці.
6. Продемонструвати результати виконаної роботи викладачу.
7. Оформити звіт про виконання лабораторної роботи, який має включати: назву теми, завдання, хід роботи, тексти програм, коментарі до них, система тестів із зазначенням до якого виду тесту належить кожен приклад.
8. Захистити звіт про виконання лабораторної роботи та відповісти на контрольні питання, які зазначені в лабораторній роботі.

Лабораторна робота №1.

Тема: Розробка системи тестів до лінійних програм.

Мета: Навчитись розробляти систему тестів до лінійних програм.

Програмне забезпечення: Pascal, Delphi 6.0, DevC++, Java.

Контрольні питання.

1. Які ви знаєте етапи розв'язання задачі з програмування?

Варіанти індивідуальних завдань:

Лабораторна робота №2.

Тема: Розробка системи тестів до лінійних програм.

Мета: Навчитись розробляти систему тестів до лінійних програм.

Програмне забезпечення: Pascal, Delphi 6.0, DevC++, Java.

Контрольні питання.

1. Налагодження.
2. Деякі типові помилки в програмах.
3. Види помилок
4. Захисне програмування.

Варіанти індивідуальних завдань:

Лабораторна робота №3.

Тема: Розробка системи тестів до програм з розгалуженням

Мета: Навчитись розробляти систему тестів до програм з розгалуженням.

Програмне забезпечення: Pascal, Delphi 6.0, DevC++, Java.

Контрольні питання.

1. Методологічні принципи налагодження програм.
2. Види тестових даних.
3. Приклади складання тестів

Варіанти індивідуальних завдань:

Лабораторна робота №4.

Тема: Розробка системи тестів до циклічних програм.

Мета: Навчитись розробляти систему тестів до циклічних програм.

Програмне забезпечення: Pascal, Delphi 6.0, DevC++, Java.

Контрольні питання.

1. Перерахуйте принципи тестування.

Варіанти індивідуальних завдань:

Лабораторна робота №5.

Тема: Розробка системи тестів до циклічних програм.

Мета: Навчитись розробляти систему тестів до циклічних програм.

Програмне забезпечення: Pascal, Delphi 6.0, DevC++, Java.

Контрольні питання.

1. Поясніть, чому тестування є деструктивним процесом.
2. Чому при тестуванні необхідно бажати, щоб програма дала збій?

Варіанти індивідуальних завдань:

Лабораторна робота №6.

Тема: Розробка системи тестів до циклічних програм.

Мета: Навчитись розробляти систему тестів до циклічних програм.

Програмне забезпечення: Pascal, Delphi 6.0, DevC++, Java.

Контрольні питання.

1. Дайте визначення терміна «помилка».
2. Наведіть класифікацію помилок по часу їх появи.

Варіанти індивідуальних завдань:

Лабораторна робота №7.

Тема: Розробка системи тестів до циклічних програм.

Мета: Навчитись розробляти систему тестів до циклічних програм.

Програмне забезпечення: Pascal, Delphi 6.0, DevC++, Java.

Контрольні питання.

1. Які помилки (в різних класифікаціях) бувають в програмах на мові C і коли вони з'являються?
2. Які мови виявляють помилки структурного набору?
3. Наведіть класифікацію помилок за ступенем порушення логіки.

Варіанти індивідуальних завдань:

Лабораторна робота №8.

Тема: Розробка системи тестів до програм з масивами.

Мета: Навчитись розробляти систему тестів до циклічних програм.

Програмне забезпечення: Pascal, Delphi 6.0, DevC++, Java.

Контрольні питання.

1. Визначте вид помилки: `if ((x > 3) && (x < 2)) ...`
2. Які типові помилки зустрічаються в програмах?
3. Деякі принципи тестування програм.

Варіанти індивідуальних завдань:

Лабораторна робота №9.

Тема: Розробка системи тестів до програм з масивами.

Мета: Навчитись розробляти систему тестів до циклічних програм.

Програмне забезпечення: Pascal, Delphi 6.0, DevC++, Java.

Контрольні питання.

1. Які етапи включає метод наскрізного перегляду програми?
2. Наведіть приклад помилки звернення до даних.
3. Наведіть приклад помилки опису даних.

4. Наведіть приклад помилки інтерфейсу.
5. Наведіть приклад помилки передачі управління.
6. У чому полягає сутність інспекції?
7. Деякі прийоми тестування.
8. Як ви розумієте надійність ПЗ?
9. Дайте визначення поняттю «тестування».
10. Покажіть і доведіть, що абсолютне тестування неможливе на конкретній програмі.

Варіанти індивідуальних завдань:

Лабораторна робота №10.

Тема: Розробка системи тестів до програм з рядковими величинами.

Мета: Навчитись розробляти систему тестів до зазначених програм.

Програмне забезпечення: Pascal, Delphi 6.0, DevC++, Java.

Контрольні питання.

1. Опишіть процес тестування методом покриття операторів.
2. Опишіть процес тестування методом покриття умов.
3. Опишіть процес тестування методом покриття рішень.
4. Опишіть процес тестування методом покриття рішень / умов.
5. Опишіть процес тестування методом комбінаторного покриття умов.
6. Опишіть процес тестування методом аналізу граничних значень.
7. Опишіть процес тестування методом еквівалентного роздроблення.
8. Опишіть процес тестування методом функціональних діаграм.
9. Опишіть процес тестування методом припущення про помилку.
10. У чому полягає метод висхідного тестування?

Варіанти індивідуальних завдань:

Лабораторна робота №11

Тема: Розробка системи тестів до створених проектів.

Мета: Навчитись розробляти систему тестів до циклічних програм.

Програмне забезпечення: Pascal, Delphi 6.0, DevC++, Java.

Створити власний проект у візуальному середовищі згідно варіанту та розробити систему тестів щодо його перевірки.

Варіанти індивідуальних завдань:

Тестові завдання

Тема 1. Основні етапи розв'язування задач з програмування

Тема 2. Тестування і налагодження програм

Тема 3. Налagodження і тестування програмних засобів

Тема 4. Тестування програми як "чорного" та "білого" ящика

Тема 5. Принципи тестування

Тема 6. Типи помилок і ручні методи тестування.

Тема 7. Стратегії білого ящика

Тема 8. Стратегії чорного ящика

Тема 9. Низхідне та висхідне тестування

Тема 10. Проектування і виконання тесту

Рекомендована література

1. Аллен И., Голуб. С и С++. Правила программирования. – М.: БИНОМ, 1996. – 272 с.
2. Вакалюк Т. А. Підготовка майбутніх учителів інформатики до розвитку логічного мислення старшокласників : дис. ... канд. пед. наук : 13.00.02 / Вакалюк Тетяна Анатоліївна. – Житомир, 2013. – 301 с.
3. Вакалюк Т.А. Підготовка майбутніх учителів інформатики до тестування програмного забезпечення // Автоматизація та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку: матеріали Всеукраїнської науково-практичної Internet-конференції. – Черкаси, 2013. – С. 275-277.
4. Вирт Н. Систематическое программирование. – М.: Мир, 1977. – 183 с.
5. Дал У., Дейкстра Э., Хоор К. Структурное программирование. – М.: Мир, 1973. – 247 с.
6. Дейкстра Э. Дисциплина программирования. – М.: Мир, 1978. – 275 с.
7. Жоголев Е.А. Технология программирования. – М., Научный мир, 2004. – 216 с.
8. Зелковиц М., Шоу А., Гэннон Дж. Принципы разработки программного обеспечения. – М.: Мир, 1982. – 368 с.
9. Зубкова Т.М. Технология разработки программного обеспечения: учебное пособие. – Оренбург: ГОУ ОГУ, 2004. – 101 с.
- 10.Иванова Г. С. Основы программирования: Учебник для вузов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. – 416 с.
- 11.Иванова Г. С. Технология программирования: Учебник для вузов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. – 320 с.
- 12.Йодан Э.. Структурное проектирование и конструирование программ. – М.: Мир, 1979. – 415 с.
- 13.Керниган Б., Пайк Р. Практика программирования. – СПб.: Невский диалект, 2001. – 381 с.
- 14.Коффман Э.Б.. Turbo Pascal. – М.: Издательский дом «Вильямс», 2002. – 896 с.
- 15.Майерс Г. Искусство тестирования программ / Пер. с англ. под ред. Б. А. Позина. – М.: Финансы и статистика, 1982. – 176 с.
- 16.Першиков В. И., Савинков В. М. Толковый словарь по информатике. – М.: Финансы и статистика, 1991. – 543 с.
- 17.Степанченко И.В. Методі тестирования программного обеспечения: учеб. Пособие /ВолгГТУ, Волгоград, 2006. – 74 с.
- 18.Тассел Д. Ван. Стиль, разработка, эффективность, отладка и испытание программ. – М.: Мир, 1985. – 332 с.
- 19.Ascoly J. et al. Code Inspection Specification.– TR–21. 630, IBM System Communication Division, Kingston, N. Y., 1976.
- 20.Boehm B.W., Software Engineering Economics, Prentice-Hall, Englewood Cliffs, NJ, 1980.

21. Brooks F.B., The mythical man-month and other essays on software engineering. Anniversary edition. University of North Carolina at Chapel Hill, 1995. Русский перевод: Брукс Ф. Мифический человеко-месяц или как создаются программные системы: Пер. с англ. – СПб.: Символ-Плюс, 1999. – 304 с.
22. Copi I. M. Introduction to Logic. New York, Macmillan, 1968.
23. Fagan M. E. Design and Code Inspections to Reduce Errors in Program Development. – IBM Systems J., 1976, 15(3), p. 182–211.
24. Freeman R. D. An Experiment in Software Development. – The Bell System Technical Journal, Special Safeguard Supplement, 1975, p. 199–209.
25. Hetzel W.C. (ed.). Program test methods, Prentice-Hall, Inc., 1973.
26. IBM Developer Works [Электронный ресурс] / Diagnosing Java Code: Designing "testable" applications. / Eric E. Allen – Электрон. дан. 2001. – <http://www-106.ibm.com/developerworks/java/library/j-diag0911.html>, свободный. – Загл. с экрана. – Яз. англ.
27. Myers G. J. A Controlled Experiment in Programm Testing and Code Walkthroughs/Inspections. – Commun. ACM, 1978, 21(9), p. 760–768.
28. Myers G. J. Composite/Structured Design. New York, Van Nostrand Reinhold, 1978.
29. Myers G. J. Software Reliability: Principles and Practices. New York, Wiley-Interscience, 1976. Русский перевод: Майерс Г. Надежность программного обеспечения. – М.: Мир, 1980.
30. Perriens M. P. An Application of Formal Inspections to Top-Down Structured Program Development. – RADCS-TR-77-212, IBM Federal System Div., Gaithersburg, Md., 1977 (NTIS AD/A-041645).
31. Shooman M. L., Bolsky M. I. Types, Distribution and Test and Correction Times for Programming Errors. – Proceedings of the 1975 International Conference on Reliable Software. New York, IEEE, 1975, p. 347–357.
32. Weinberg G. M. The Psychology of Computer Programming. New York, Van Nostrand Reinhold, 1971.
33. Weinberg G. M. The Psychology of Computer Programming. New York, Van Nostrand Reinhold, 1971.
34. Yourdon E. Techniques of program structure and design. Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1975. Русский перевод: Йодан Э. Структурное проектирование и конструирование программ. – М.: Мир, 1979. – 416 с.

Технології тестування програм

Навчальне видання

ВАКАЛЮК Тетяна Анатоліївна

Технології тестування програм

*Навчально-методичний посібник для студентів напрямку
6.040302 Інформатика*.*

Надруковано з оригінал-макета автора

Підписано до друку 25.10.13. Формат 60x90/16. Папір офсетний.

Гарнітура Times New Roman. Друк різнографічний.

Ум. друк. арк. 5,6. Обл. вид. арк. 5. Наклад 300. Зам. 825.

Видавець і виготовлювач

Видавництво Житомирського державного університету імені Івана Франка

м. Житомир, вул. Велика Бердичівська, 40

Свідоцтво суб'єкта видавничої справи:

серія ЖТ №10 від 07.12.04 р.

електронна пошта (E-mail): zu@zu.edu.ua