

ПІДГОТОВКА МАЙБУТНІХ УЧИТЕЛІВ ІНФОРМАТИКИ ДО ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

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 напрямом 6.040302 "Інформатика". Метою даної дисципліни є допомогти студентам оволодіти теоретичними знаннями та практичними навиками роботи з управлінням якістю програмного забезпечення на різних етапах життєвого циклу, тестування з метою створення корисних і працездатних програмних продуктів.

Програмні помилки, як правило поділяються на три види: *синтаксична помилка* – неправильне використання синтаксичних конструкцій, або помилка в написанні зарезервованих слів. Ці помилки виявляти найпростіше, адже компілятор сам виявить їх і вкаже на них; *семантична помилка* – помилка у програмі, яка пов'язана з неправильним змістом дій та використанням недопустимих значень величин (наприклад, помилки даних: символічні замість числових, ділення на 0, корінь з від'ємного числа тощо); *логічна помилка* – порушення логіки програми, яке призводить до неправильного результату. Подібні помилки ховаються в алгоритмах і потребують ретельного аналізу та всебічного тестування.

Для налагодження найчастіше використовують покрокове виконання програми, що забезпечує слідкування за значеннями змінних на різних етапах виконання програми.

Тому для зменшення ймовірності виникнення помилок, використовується *захисне програмування*, що полягає у стилі написання програм, при якому помилки, які з'являються, легко виявляються та ідентифікуються програмістом.

Найпростіший метод використання захисного програмування полягає у тому, що при написанні програми потрібно передбачити опрацювання ситуацій, які не можуть статись ні за яких обставин. Для цього необхідно:

1. Перевіряти тип вхідних даних. Контролювати літерні поля, щоб переконатися, що вони не містять цифрових даних. Перевіряти цифрові поля на відсутність в них літерних даних.

2. Здійснювати перевірку області значень змінних, щоб упевнитися, наприклад, що додатні величини завжди додатні.

3. Виконувати контроль правдоподібності значень змінних, які не повинні перевищувати деяких значень.

4. Контролювати підсумки обчислень шляхом введення всюди, де це можливо, перекресних підсумків та контрольних сум.

Тестування – це виконання комплексу завдань для перевірки на правильну працездатність програми. Тестування може виявити факт наявності помилки, а налагодження виявляє причину помилки, так що ці два етапи розробки програми «перекриваються».

Один із корисних принципів тестування полягає в тому, щоб кожний оператор програми був випробуваний хоча б один раз. Якість тестування визначається значним чином не кількістю тестових прогонів. Головне, щоб кожний тестовий прогін контролював би щось таке, що не було перевірено в попередніх прогонах.

Наведемо основні принципи тестування програм:

- 1) Використання принципу *захисного програмування* (див. вище).
- 2) *Тестування граничних умов*: в умовному виразі необхідно переконатися, що розгалуження виконується правильно; при написанні циклу потрібно передбачити перевірку того, чи тіло циклу буде виконано потрібну кількість разів.
- 3) *Аналіз результатів тестування*: для порівняння обчислити результат іншим способом.
- 4) *Тестування окремих блоків незалежно один від одного*. Потрібно враховувати проміжні результати [1].

Список використаних джерел

1. Майерс Г. Искусство тестирования программ / Пер. с англ. под ред. Б. А. Позина. – М.: Финансы и статистика, 1982. – 176 с.