

Лабораторна робота № 6

Складання звіту про помилки (4 години)

Мета: набуття практичних навичок застосування різних технік ручного тестування та опису виявлених дефектів у вигляді баг-репортів.

Теоретична частина

Функціональне тестування – це один з видів тестування, спрямованого на перевірку відповідностей функціональних вимог програмного забезпечення (ПЗ) його реальним характеристикам. Основним завданням функціонального тестування є підтвердження того, що програмний продукт, який розробляється, має весь функціонал, необхідний для замовника. Залежно від мети, функціональне тестування може проводитися: на підставі функціональних вимог, зазначених у специфікації вимог.

При цьому для тестування створюються тестові випадки (testcases), складання яких ураховує пріоритетність функцій програмного забезпечення, які необхідно покрити тестами. Таким чином можна переконатися в тому, що всі функції розроблюваного продукту працюють коректно з використанням різних типів вхідних даних, їх комбінацій, кількості, на підставі бізнес-процесів, які має забезпечити додаток. У такому разі важливим є не так працездатність окремих функцій ПЗ, як коректність виконуваних операцій, з огляду на сценарії використання системи. Отже тестування в такому разі ґрунтуватиметься на варіантах використання системи (usecases). Описані вище аспекти реалізуються за допомогою таких напрямів і рівнів тестування:

- модульне (компонентне);
- інтеграційне;
- системне;
- регресійне;
- приймальне.

Нефункціональне тестування – це тестування властивостей, які не належать до функціональності системи і визначаються нефункціональними вимогами, які характеризують продукт з таких сторін:

- надійність (реакція системи на непередбачені ситуації);
- продуктивність (працездатність системи під різними навантаженнями);
- зручність (зручність роботи з додатком з огляду на користувача);
- масштабованість (вимоги до горизонтального або вертикального масштабування застосунку);
- безпека (захищеність призначених для користувача даних);
- портируемість (можливість перенесення на різні платформи).

Ці властивості системи можна досліджувати, використовуючи такі види тестування:

Тестування установки (Installation testing) – перевірка успішності установки додатка, його налаштування та видалення, що зменшує ризики втрати призначених для користувача даних, втрати працездатності програми та ін.

Тестування зручності використання (Usability testing) – характеризує систему щодо зручності використання кінцевим користувачем.

Конфігураційне тестування – дослідження працездатності програмної системи в умовах різних програмних конфігурацій.

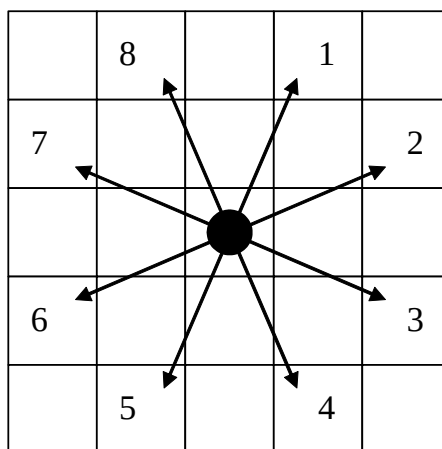
Тестування на відмову та відновлення (Failover and Recovery Testing) – дослідження програмної системи щодо відновлення після помилок, збоїв. Оцінювання реакції захисних властивостей програми.

Структура звіту з лабораторної роботи та порядок роботи

1. Завантажити десктопну програму «KNIGHT» з матеріалів ПНС.
2. Інсталиувати програму «KNIGHT» на свій комп'ютер.
3. Занотувати у звіт короткий опис функціоналу цієї програми (повний опис принципу роботи, доступний за допомогою меню «Help»).
4. Провести тестування «KNIGHT» з використанням різних технік, знайти якомога більше багів.
5. Усі знайдені баги подати у вигляді баг-репортів у звіті.

1. Опис об'єкта тестування програмного продукту «KNIGHT»

Завдання програми полягає в тому, щоб знайти всі варіанти обходу дошки конем з покриттям максимально можливого числа клітин із заданою позиції. Принцип роботи програми: вибираються гілки і проводиться їх перевірка. Що таке гілка? Це послідовність ходів, кожному з яких присвоєно порядковий номер у відповідності з наступним малюнком:



Елементи управління головної форми

Кнопка Start - запускає обчислення.

Choose Decision combobox - призначений для вибору рішення, яке буде відтворюватись.

Кнопка Play - запускає програвання рішення.

Швидкість програвання регулюється трекбаром Slow Play - Fast Play.

Прапорець Animate управляє видом програвання (з анімацією або без). Він дійсний тільки для режиму показу Knight View Mode.

Меню

File | Select Start Position - переводить програму в режим вибору старто-вої позиції. Курсор приймає вид хрестика і при кліці на дошці старто-вая позиція переміщається в відповідну клітинку.

File | Go - запускає обчислення.

File | Clear - очищає дошку.

File | Play Decisions - запускає програвання рішення.

File | Options ... - викликає Options dialog.

File | Exit - вихід з програми.

Help | About - викликає About dialog.

Опції (Options dialog)

Lines Number - розмір збоку дошки (3..10).

Cell Side - розмір сторони чарунки у пікселях (30..100).

Delay Between Moves, ms - пауза між ходами при розрахунку (0..5000). Використовується, щоб легко можна було зрозуміти принцип роботи програми. Логічно застосовувати разом з обраної Show Graphics during Calculations.

Show Branches Detailed Info - показувати детальну інформацію про гілки обчислень.

Show Graphics during Calculations - показувати графіку під час обчислень.

Write Branches To File - записувати гілки обчислень в файл.

Write Max Branches To File - записувати рішення в файл.

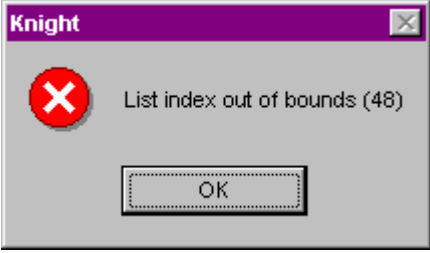
View Mode - режим відображення

Numbers - кольорові квадратики з цифрами;

Knights - кольорові фігурки коня з цифрами.

2. Зразок написання звіту про помилку (bug report)

Заголовок	Knight: поле введення Choose decision: повідомлення про помилку
Категорія <i>Функціональна, інтерфейсу, текстова</i>	Функціональна
Повторюваність <i>Завжди, часто, рідко, не можна повторити</i>	Завжди
Пріоритет <i>Високий, середній, низький</i>	Високий

Опис <i>Кроки для повторення</i>	1. Запустити програму 2. У поле Choose Decision вибрати - 48 3. Натиснути кнопку «Play» 4. Виникає повідомлення про помилку 
Зауваження	Програма повинна працювати без цього повідомлення