

Лекція 3. Серія тестів

Перший цикл тестування.

Отримуємо програми і наступний опис її функціонування:

Призначення програми – скласти два введені користувачем числа. В кожному з чисел може бути одна або дві цифри. Програма відображує введені числа і після цього виводить їх суму. Введення кожного числа закінчується натисненням клавіші Ентер. Запускається програма з допомогою команди ADDER (adder.exe).

КРОК 1. Простий і найбільш очевидний тест

В програмах, представлених для першого формального тестування, часто одразу виникає збій. Тому варто виконати найпростіший тест.

Дія	Що відбувається
Вводимо ADDER і натискаємо <Ентер>	Вврхе екрану з'являється знак питання. Курсор мигає
Натискаємо 2	За знаком питання з'являється цифра 2
Натискаємо <Ентер>	В наступному рядку з'являється знак питання
Натискаємо 3	За другим знаком питання з'являється цифра 3
Натискаємо <Ентер>	В третьому рядку з'являється цифра 5. На декілька рядків нижче з'являється ще один знак питання

Звіт про проблеми першого тесту.

- ▣ *Помилка проектування.* Нема вказівок на те з якою саме програмою ви працюєте
- ▣ *Помилка проектування.* На екрані нема жодних інструкцій. Звідки знати що робити? Відобразити інструкцію на екрані нескладно, а друкована документація може загубитись.
- ▣ *Помилка проектування.* Як зупинити виконання програми? Ця інструкція має бути на екрані.
- ▣ *Помилка кодування.* Цифра 5 виведена в стороні від доданків.

Обов'язково необхідно представляти детальний звіт про кожну проблему. Можна згрупувати помилки в один звіт, але цього робити не варто.

Документ “звіт про проблему”

КРОК 2. Що ще має бути протестовано

Приклад серії тестів.

Вхідні дані	Очікуваний результат	Замітки
99 + 99	198	Пара найбільших чисел які може скласти програма
-99 + -99	-198	В документації не сказано, що неможна <u>скаладати</u> від'ємні числа
99 + -14	85	Перше велике число може вплинути на інтерпретацію системою другого
-38 + 99	61	Перевіримо додавання від'ємного і додатного чисел
56 + 99	155	Друге велике число може вплинути на інтерпретацію системою першого
9 + 9	18	9 – найбільше число з однієї цифри
0 + 0	0	Програми часто злітають на нулях
0 + 23	23	Програма може неправильно обробляти 0 і тому його треба перевірити в якості обох доданків
-78 + 0	-78	

Загальна кількість тестів 39601. Допустимий діапазон (-99,99) - усього 199. Отже, 199^2 . Це без врахування будь-яких складних дій користувача (наприклад натиснення бекспейсів і делете). Ефективніше усього перевіряти граничні умови. Якщо для двох тестів очікується однаковий результат, то тести належать до одного класу. В нашому випадку 81 тест відноситься до класу «пара однозначних додаткових чисел».

Для виконання слід вибирати ті тести з класу, на яких ймовірноше усього може відбутись збій програми.

Класом можна назвати групу значень, які програма обробляє однаковим чином. А граничними значеннями класу є ті вхідні дані, на яких програма міняє свою поведінку. Границю завжди слід перевіряти з обох боків.

Програмісти часто переконуються, що критичний фрагмент коду працює на одному із значень і забувають це зробити на другому.

КРОК 3. Перевірка недопустимих значень.

Програму тестують тому, що вона може не працювати.

На цьому етапі слід перевірити недопустимі значення.

В нашому випадку:

- ▣ Додаткові числа і нулі обробляються правильно.
- ▣ Не працює жоден тест з від'ємними числами. Після вводу другої цифри комп зависає. Очевидно програма не очікує вводу від'ємних чисел

КРОК 4. Трохи тестування в режимі «вільного польоту»

<i>Тест</i>	<i>Чи цікавий тест</i>	<i>Зауваження</i>
100 + 100	Гранична умова: числа більша за допустимий максимум	Програма прийняла 10. Коли ви ввели другий нуль, програма повела себе так, ніби була натиснутий ентер. Те саме з другим числом. В результаті – сума 20
<Enter> + <Enter>	Відсутність введених даних	Коли ви натиснули Ентер, програма надрукувала 10, останнє введене вами число
123456 + 0	Побільше цифр	Програма прийняла перші дві цифри як і у випадку з цифрою 100. В майбутній версії програма має працювати з більшими числами. Як тоді вона має реагувати на такі ситуації?
1,2 + 5	Десятковий знак	Реакція на десятковий знак така ж як і на Ентер
A + b	Недопустимі символи	Ви натиснули Ентер – після <A> програма зависла
<Ctrl>+<A> + <Ctrl>+, <F1>+<Esc>	Керуючі символи і функціональні клавіші часто є джерелами пролем	Для усіх комбінацій клавіш програма виводить графічні символи, потім після натиснення на Ентер зависає

КРОК 5. Підсумки про недоліки програми

- ▣ У програми дуже обмежений інтерфейс
- ▣ Програма не працює з від'ємними числами. Найбільша обчислювана сума – 198, найменша – 0.
- ▣ Третій ввідний символ програма інтерпретує як натиснутий Ентер
- ▣ Поки не натиснуто Ентер будь-які символи сприймаються як допустимі
- ▣ Програма не перевіряє чи дійсно введена цифра.

Якщо програміст не зовсім некомпетентний, то для таких результатів має бути причина. Скоріш за все програміст намагався зробити програму якомога меншою за розміром або швидкою.

Підсумки першого циклу тестування

Оскільки програма пройшла простий тест, була розроблена серія формальних тестів для перевірки роботи з допустимими даними. Ці тести будуть використані і далі.

Оскільки частину перевірок програма не пройшла, на планування подальших тестів поки доцільно не витратити час. Замість цього було проведено ряд неформальних експериментів і виявлено, що програма дуже нестабільна.

До написаних зауважень слід повернутись при наступному тестуванні програми.

ДРУГИЙ цикл тестування.

Програміст повідомив, що швидкість роботи є дуже важливою, а об'єм коду не має значення. Програміст поставив свої резолюції.

Помилка	Резолюція
На екрані нема назви програми На екрані нема інструкцій	Не буде виправлена Не буде виправлена. Вивід інструкцій уповільнить роботу програми
Як зупинити програму?	Виправлена. На екрані відображається підказка «Для виходу натисніть <Ctrl>+<C>»
Сума 5 виводиться в стороні від доданків	Виправлена
Програма зависає на від'ємних числах	Виправлена. Програма буде складати і від'ємні числа
Програма інтерпретує третій введений символ як Ентер	В стадії розробки (ще не виправлена)
Збій при вводі нечислових даних	Не проблема. Коментар: не вводьте нечислові дані
Збій при вводі керуючих символів	Не проблема
Збій при натисненні функціональних клавіш	Не проблема

КРОК 1. Уважно прочитайте резолюції програміста і визначте, що треба робити, а що ні.

Із резолюцій чітко видно які тести більше проводити не треба, а які будуть замінені новими. Резолюції програміста є ключем до створення нової серії тестів

КРОК 2. Проаналізуйте коментарі до помилок, які не будуть виправлені. Можливо необхідно провести додаткове тестування.

Для того, щоб добитись виправлення помилки, необхідно продемонструвати ситуацію в якій її появлення абсолютно недопустиме. Програма зависає при введенні будь-якого недопустимого символу – необхідно продемонструвати програмісту, що це неправильно.

КРОК 3. Передивіться записи першого циклу, додайте нові зауваження і переходьте до тестування.

Слід провести серію уже проведених тестів, для того щоб переконатись, що програма їх виконує.

Після їх проведення ви помітили, що програма відображає підказку «Для виходу натисніть <Ctrl>+<C>» після кожної операції додавання. Можна скласти наступне зауваження:

Помилка проектування. На вивід на екран підказки тратиться зайвий час. Можна просто написати внизу екрана цю підказку перед початком роботи. Заодно і додати заголовок програми і короткі інструкції.

Оскільки програма продовжує зависати при введенні недопустимих символів, то слід описати проблему ще раз.

Можливий варіант виправлення – перевіряти кожний введений символ.

Недопустимі ігнорувати і виводити повідомлення про помилку.

Головне – указати, що проблема серйозна.

Хороший тестер не той, хто виявить найбільше помилок і не той, хто примусить збентежитись навіть самого першокласного програміста. Кращим є той, хто досягне виправлення найбільшої кількості помилок.