

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Український державний університет науки і технологій**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №1  
з дисципліни «Якість програмного забезпечення та тестування»**

**на тему:**

**«**Тестування методами білої скриньки.»

Виконав:

студент гр. ПЗ2011

Проценко Р.О.

Прийняла:

Стаднік А. В.

Дніпро, 2023

**Тема.** Тестування методами білої скриньки.

**Мета.** Отримати практичні навички тестування методами білої скриньки.

**Завдання**

Підготувати дві чи більше функцій для тестування. Функції повинні містити не менше шести керуючих конструкцій (розгалужень і циклів). Зазначені структури мають бути вкладеними. Виконати тестування функцій методами білої скриньки.

**Специфікація функції №1**

Функція здійснює множення двох матриць між собою.

Функція приймає на вхід у явному виді:

* a – двовимірний масив чисел з плаваючою комою.
* b - двовимірний масив чисел з плаваючою комою.

У не явному вигляді функція повертає виключення пов'язані з невідповідністю розмірів матриць.

У явному повертає двовимірний масив чисел з плаваючою комою, результат множення двох матриць.

**Текст функції №1**

public static double[][] dot(double[][] a, double[][] b) {  
 int rowA = a.length, rowB = b.length; \\1  
 int colA = a[0].length, colB = b[0].length; \\2  
  
 if (colA != rowB) { \\3  
 throw new IllegalArgumentException("Matrix dimensions are not compatible for multiplication."); \\4  
 }  
  
 if (colB \* rowB <= 1 || rowA \* colA <= 1) { \\5  
 throw new IllegalArgumentException("The function received a scalar instead of a matrix"); \\6  
 }  
  
 double[][] result = new double[rowA][colB]; \\7  
  
 for (int i = 0; i < rowA; i++) { \\8  
 for (int j = 0; j < colB; j++) { \\9  
 double sum = 0.0; \\10  
 for (int k = 0; k < colA; k++) { \\11  
 sum += a[i][k] \* b[k][j]; \\12  
 }  
 result[i][j] = sum; \\13  
 }  
 }  
  
 return result; \\14  
}

**Тестування**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Назва | Вхідні дані | Очікувані результати |
| 1 | Множення матриць з правильними розмірами. | а = 1, 2 , 3  4, 5, 6  b = 7, 8  9, 10  11, 12 | Result = 58, 64  139, 154 |
| 2 | Множення матриць з неправильними розмірами. | а = 1, 2 , 3  4, 5, 6  b = 7, 8  9, 10 | IllegalArgumentException  ("Matrix dimensions are not compatible for multiplication.") |
| 3 | Множення скаляра на скаляр. | а = 1  b = 7 | IllegalArgumentException  ("The function received a scalar instead of a matrix") |
| 4 | Множення вектора на скаляр. | а = 1, 2, 3, 4, 5, 6  b = 7 | IllegalArgumentException  ("The function received a scalar instead of a matrix") |
| 5 | Множення вектора на скаляр. | а = 1  b = 7, 8, 9 | IllegalArgumentException  ("The function received a scalar instead of a matrix") |

**Таблиця покриття операторів**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Оператор  Тест | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 1 | \* | \* | \* |  | \* |  | \* | \* | \* | \* | \* | \* | \* | \* |
| 2 | \* | \* | \* | \* |  |  |  |  |  |  |  |  |  |  |
| 3 | \* | \* | \* |  | \* | \* |  |  |  |  |  |  |  |  |
| 4 | \* | \* | \* |  | \* | \* |  |  |  |  |  |  |  |  |

**Таблиця покриття умов**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Умова  Тест | colA != rowB | | colB \* rowB <= 1 | | rowA \* colA <= 1 | | i < rowA | | j < colB | | k < colA | |
| + | - | + | - | + | - | + | - | + | - | + | - |
| 1 | \* |  |  | \* |  | \* | \* | \* | \* | \* | \* | \* |
| 2 |  | \* |  | \* |  | \* |  |  |  |  |  |  |
| 3 | \* |  | \* |  | \* |  |  |  |  |  |  |  |
| 4 | \* |  | \* |  | \* |  |  |  |  |  |  |  |

**Таблиця покриття рішень**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Умова  Тест | colA != rowB | | colB \* rowB <= 1 || rowA \* colA <= 1 | | i < rowA | | j < colB | | k < colA | |
| + | - | + | - | + | - | + | - | + | - |
| 1 |  | \* |  | \* | \* | \* | \* | \* | \* | \* |
| 2 | \* |  |  |  |  |  |  |  |  |  |
| 3 |  | \* | \* |  |  |  |  |  |  |  |
| 4 |  | \* | \* |  |  |  |  |  |  |  |

**Таблиця комбінаторного покриття умов**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Умова  Тест | colB \* rowB <= 1 || rowA \* colA <= 1 | | | |
| + + | +- | -+ | -- |
| 1 |  |  |  | \* |
| 2 |  |  |  | \* |
| 3 | \* |  |  |  |
| 4 |  | \* |  |  |
| 5 |  |  | \* |  |

**Специфікація функції №2**

Функція здійснює поелементне двох матриц.

Функція приймає на вхід у явному виді:

* a – двовимірний масив чисел з плаваючою комою.
* b - двовимірний масив чисел з плаваючою комою.

У не явному вигляді функція повертає виключення пов'язані з невідповідністю розмірів матриць.

У явному повертає двовимірний масив чисел з плаваючою комою, результат додавання двох матриць.

**Текст функції №2**

public static double[][] add(double[][] a, double[][] b) {  
 if (a.length == b.length && a[0].length == b[0].length) { //1  
 throw new IllegalArgumentException("Matrix dimensions must be same.");//2  
 }  
  
 double[][] result = new double[a.length][a[0].length];//3  
  
 for (int i = 0; i < a.length; i++) {//4  
 for (int j = 0; j < a[0].length; j++) {//5  
 result[i][j] = a[i][j] + b[i][j];//6  
 }  
 }  
  
 return result;//7  
}

**Тестування**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Назва | Вхідні дані | Очікувані результати |
| 1 | Додавання матриць | а = 1, 2  4, 5  b = 7, 8  9, 10 | Result = 8, 10  13, 15 |
| 2 | Додавання матриць з неправильними розмірами. | а = 1, 2 , 3  4, 5, 6  b = 7, 8  9, 10 | IllegalArgumentException("Matrix dimensions must be same.") |
| 3 | Множення скаляра на скаляр. | а = 1, 2  b = 7, 8, 9  10, 7, 3 | IllegalArgumentException("Matrix dimensions must be same.") |
| 4 | Множення вектора на скаляр. | а = 1, 2  3, 4  5, 6  b = 7, 2 | IllegalArgumentException("Matrix dimensions must be same.") |

**Таблиця покриття операторів**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Оператор  Тест | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | \* |  | \* | \* | \* | \* | \* |
| 2 | \* | \* |  |  |  |  |  |
| 3 | \* | \* |  |  |  |  |  |
| 4 | \* | \* |  |  |  |  |  |

**Таблиця покриття умов**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Умова  Тест | a.length == b.length | | a[0].length == b[0].length | | i < a.length | | j < a[0].length | |
| + | - | + | - | + | - | + | - |
| 1 | \* |  | \* |  | \* | \* | \* | \* |
| 2 | \* |  |  | \* |  |  |  |  |
| 3 |  | **\*** |  | **\*** |  |  |  |  |
| 4 |  | **\*** | \* |  |  |  |  |  |

**Таблиця покриття рішень**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Умова  Тест | a.length == b.length && a[0].length == b[0].length | | i < a.length | | j < a[0].length | |
| + | - | + | - | + | - |
| 1 | \* |  | \* | \* | \* | \* |
| 2 |  | \* |  |  |  |  |
| 3 |  | \* |  |  |  |  |
| 4 |  | \* |  |  |  |  |

**Таблиця комбінаторного покриття умов**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Умова  Тест | a.length == b.length && a[0].length == b[0].length | | | |
| + + | +- | -+ | -- |
| 1 | \* |  |  |  |
| 2 |  | \* |  |  |
| 3 |  |  |  | \* |
| 4 |  |  | \* |  |

**Висновки**

Тестування методами білої скриньки - це важлива частина процесу розробки програмного забезпечення. Воно дозволяє перевірити внутрішню структуру програми, включаючи модулі, функції та гілки коду. Головною метою цього виду тестування є забезпечення відповідності програми вимогам та забезпечення її стабільної роботи.

Тестування білої скриньки проводиться паралельно з іншими методами тестування для забезпечення повноти оцінки програмного продукту. Під час цього тестування, тестувальники мають доступ до внутрішньої структури програми, що дозволяє створювати тестові випадки на основі коду.

Під час тестування методами білої скриньки можуть бути виявлені різні дефекти, такі як логічні помилки та проблеми з безпекою. В результаті, це сприяє покращенню якості та надійності програмного продукту, що є важливим завданням в розробці програмного забезпечення.